

**Chapter
2**

XEDIT

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives**Chapter 2: XEDIT**

You will learn:

- C Invoking/leaving XEDIT.
- C Editing.
- C Prefix subcommands.
- C Scrolling subcommands.
- C Copying and moving.
- C Block prefix subcommands.
- C GET and PUT.
- C REPEATING commands.
- C HELP.
- C SET.
- C TRUNC and VERIFY.
- C WRAP and STAY.
- C Multiple files.
- C Split screens.
- C XEDIT and EDGAR.
- C XEDIT and ICCF.

1 Purpose and Function

XEDIT is the full screen editor supplied with VM/SP. It supercedes the previous editors, EDGAR and EDIT, having all the best features of both along with its own additional subcommands and features, including:

- C Macro facility
- C Screen tailoring
- C Word processing features
- C Profile macro
- C Multiple files
- C Multiple Screens
- C Screen or Typewriter mode

XEDIT usually runs in screen mode, but may also be used in typewriter mode, which is useful when using XEDIT in EXEC procedures. When in screen mode data may be changed on the screen, or manipulated and changed by using subcommands, prefix subcommands and PF keys.

XEDIT can edit multiple files, and is able to have several logical displays on a screen, showing several files, or different parts of the same file. The screen may be split horizontally (and, from Release 3, vertically).

The format of the screen is tailorable by the user, and this can be done by a PROFILE macro which is executed when XEDIT is initiated.

This course segment is in two parts; the first describes how XEDIT is started, terminated, how new files are created and describes the screen format and the second describes the basic editing subcommands.

2 XEDIT: Invoking

Issuing this command will edit a file:

XEDIT filename filetype filemode

- C If the filemode is omitted, all disks are searched in the normal way.
- C If no file is found, a new one is created with mode A.

XEDIT is often invoked from FLIST or FILELIST, in which case the cursor is positioned next to the required file and the XEDIT PF key pressed (usually PF4 for FLIST and PF11 for FILELIST).

If a CMS file called PROFILE XEDIT is found when XEDIT is invoked, it is executed. There is normally a PROFILE XEDIT on one of the system disks.

- C It can be copied into the A disk and tailored to the screen and predefine some of the SET values.
- C It is also used to set up default LRECL, TRUNC, TAB, VERIFY etc. values for filetypes not set up as standard.

3 Standard XEDIT Screen

The standard XEDIT screen looks like this:

```

PROFILE EXEC  A1 V 130 TRUNC=130 SIZE LINE=0 COLUMN=1

===== * * * TOP OF FILE * * *
      ...+....1....+....2....+....3....+....4....+....5....6..  ..+....7....
===== &CONTROL OFF
===== * SET UP PF KEYS
===== CP SET PF1 IMM HELP HELP
===== CP SET PF3 IMM #CP DISC
===== CP SET PF9 IMM #CP Q R ALL
===== CP SET PF12 IMM FLIST
===== CP SET PF8 DEPRINT
===== * SET UP PUNCH
===== CP SPOOL PUN VSEIPO
====> -

                X E D I T  1  F I L E

```

The current line is in the center of the screen. Subcommands may be entered in the command area, prefix subcommands in the prefix area (denoted by = signs). The command area also displays messages. The information along the top of the screen displays information about the file, such as filename, filetype, filemode, its record format, etc. This information is protected.

When XEDIT is invoked, it runs the PROFILE macro. The personal profile can be tailored or the installation PROFILE may be used. Commands to alter the screen layout may be included in the installation PROFILE, accordingly, the screen layout may differ to that shown above.

Records (or lines) which are longer than the display area will be displayed on more than one line.

4 XEDIT: Leaving

The following subcommands are used to leave the XEDIT environment.

FILE	Saves the file with all its changes.
QUIT or PF3	Does not write the file to disk. If it has been changed there will be an error message, and QQUIT should be used.
QQUIT	Does not write the file to disk, therefore the changes will be lost.

Upon leaving XEDIT, the operating environment will return to CMS or FLIST.

5 Editing New Files

If the file being XEDITed does not exist, then a new one is created. The input subcommand INPUT is used for entering data into a new file. The screen goes into input mode, and lines of data can be keyed in. Upon reaching the end of a screen, hit the enter key and the screen scrolls forward so more data may be entered.

In order to leave input mode, hit the ENTER key twice and then enter FILE to write the file to disk.

Input mode may also be entered with an existing file. All data input goes after the line that was current when the INPUT subcommand was issued.

6 SAVE

When creating a large file with a large amount of data, it will be a good practice to save the data occasionally by leaving input mode and entering the SAVE command. This will write the file to disk without leaving XEDIT.



7 Editing: Basic

Most editing can be done using only a few subcommands.

In order to make changes to the file on the screen move the cursor to the correct position and overwrite the changes. The terminal's delete and insert keys may be used.

Prior to inserting characters, it is necessary to change the trailing blanks to nulls by pressing the PA2 key. Changes (and subcommands) are transmitted when the ENTER key is pressed. The cursor often remains within the body of the file or screen, and does not always return to the command line. Hitting ENTER without entering any commands will cause the cursor to return to the command line.

The CLEAR key will cause the screen to be redisplayed as it was after the ENTER key was last pressed.

8 Prefix Subcommands

Prefix subcommands are entered in the prefix area.

Several subcommands may be entered at once.

A	Add a blank line after this one.
D	Delete a line.
"	Duplicate this line.
/	Make this the current line.
E	Extend this line onto the next physical line.

The prefix subcommands A, D and " may be qualified by a numeric value, so that A3 means add three lines, D6 means delete six lines, and so on.

9 Scrolling Subcommands

The following XEDIT subcommands are used to move through the file:

FORWARD n	Scrolls forward n screens. The default is one screen. PF8 is set to FORWARD.
BACKWARD n	Scrolls backward n screens. The default is one. PF7 is set to BACKWARD.
UP n	Moves the line pointer up (towards Top of File) n lines. Default is one.
DOWN n	Moves the line pointer down (towards End of File) n lines. Default is one.
TOP	Move to the top of the file.
BOTTOM	Move to the end of the file.

10 Targets

Some subcommands, such as locate and change, use targets to terminate their operation. For a locate subcommand for example, the target is usually a string. The usual form of a target is a string, delimited by slashes, e.g.

```
/EXEC/
```

The second delimiter is optional, so:

```
/EXEC
```

has the same meaning.

Targets are used normally with LOCATE subcommand, e.g.

```
LOCATE/EXEC
```

This searches the file for the first occurrence in the file of the string EXEC, and moves it to the current line. The LOCATE subcommand itself may be omitted, so:

```
/EXEC
```

has the same effect.

The search may be done backwards, by specifying a negative target,

```
-/EXEC
```

The target:

```
-/EXEC
```

specifies the target as the first line NOT containing EXEC.

The target:

```
/EXEC*/PAY
```

specifies the first line containing either EXEC or PAY.

Numeric targets may also be used.

5	Refers to the fifth line from the current one. The subcommand 5 is equivalent to DOWN 5.
-5	Refers to the fifth line preceding the current one.
:35	Refers to an absolute line number. This will cause the thirty-fifth line of the file to become the current one. The target * specifies the target as End of File.

11 CHANGE Subcommand

The change subcommand is usually used to change several occurrences of a string to another one.

CHANGE /old/new/ target p q

old	String to be checked.
new	What it is to be changed to.
target	Specifies where the change is to stop. No changes are made to the line containing the target. An * indicates that changes are to be made to EOF.
p	Number of times in each line the string is to be changed. * specifies every occurrence.
q	Occurrence of the string in the line where the change must start.

The default values for target, p, and q are 1.

Examples:

C /EDGAR/XEDIT	Will affect the first occurrence in the current line.
C /EDGAR/XEDIT/1 1 1	Has the same effect as the previous example.
C /REAL/VIRTUAL/ **	Changes all occurrences from current line to end of file.
C /VM/CP/ /VM COMMANDS/ *	Changes "VM" to "CP" until the line containing VM COMMANDS is reached (but does not change the line containing the target).
C /PGM=// **	Deletes all occurrences of "PGM=".
C /VIRTUAL/REAL/ 1 1 2	Changes the second occurrences of "VIRTUAL" on the current line to "REAL".

If the string to be changed contains a slash, then use another delimiter. In order to change VM/370 to VM/SP use the subcommand:

C !VM/370!VM/SP! **

XEDIT takes the first non-blank character after the change subcommand as the delimiter.

12 Selective Changes

It is possible to go through the file using PF keys to make selective changes. PF keys 5 and 6 are set by XEDIT for this purpose.

Key the change subcommands into the command line (but do not hit ENTER), then hit PF5. This will place the cursor under the next occurrence of the old string. Then hit PF6 to change it, or hit PF5 to leave it as it is and look for the next occurrence.

13 COPYING and MOVING

XEDIT has several prefix subcommands which are useful for copying or moving sections of a file.

The subcommands are:

C	copy
M	move
F	following
P	preceding

Use the C or M subcommands to denote which line should be copied or moved, and the F or P subcommands to show where it should go.

Example:

```
===== line one
===== line two
===== line three
===== line four
===== line five
===== line six
```

results in:

```
===== line one
===== line two
===== line three
===== line four
===== line five
===== line six
```

The C or M subcommands can be qualified by numeric values to copy or move several lines.

It is possible to key the copy or move subcommand and then scroll through the file before entering the P or F subcommand.

14 BLOCK PREFIX Subcommands

Some prefix subcommands can be used to act on blocks of data.

These are the Copy, Move, Delete and duplicate (") subcommands.

CC	block copy
DD	block delete
MM	block move
"	block duplicate

In order to delete a block of lines use the DD subcommand on the first and last lines of the block, e.g.

```
===== line one  
==DD= line two  
===== line three  
===== line four  
DD=== line five  
===== line six
```

results in:

```
===== line one  
===== line six
```

These blocks may start and end on different screens.

If you want to cancel a pending block operation, use the RESET subcommand.

15 FIND

The FIND subcommand will search for a line that starts with the given string.

FIND text

text	Is a string of characters. It is separated from the FIND subcommand by one blank. It is not delimited by slashes or quotes. Special characters may be included. If a blank is to be included, use the underscore character. A blank is used to represent a "wild" character.
------	--

Examples:

FIND USER_MAINT

FIND /*

16 SHIFT

The SHIFT subcommand moves data to the left or right. Truncation of data is likely to occur.

```
SHIFT LEFT  n target
           RIGHT
```

LEFT	Specifies data is to be shifted to the left.
RIGHT	Specifies data is to be shifted right.
n	Is the number of columns the data will be shifted.
target	Tells XEDIT the number of lines to shift.

Example:

```
SHIFT RIGHT 5 *
SH LE 10 /PAYROLL
```

17 RIGHT

Data can be viewed on a file (e.g. with a record length greater than 72 or 80) that is normally not visible on a 80 column screen. The RIGHT and LEFT commands provide this capability.

RIGHT provides the capability for viewing data to the right of the data on the screen.

RIGHT n

n	Number of columns the data will be moved to the right.
---	--

Example:

RIGHT 20

18 LEFT

LEFT provides the capability for viewing data that is to the left of the data on the screen.

LEFT n

n	Number of columns the data will be moved to the right.
---	--

Example:

LEF 15

These two commands are cumulative;

RIGHT 20 followed with LEFT 15 and RIGHT 5 will result in the displayed data being moved by 10 columns.

In order to restore the original display, use RIGHT 0 or LEFT 0.

19 DELETE

The DELETE subcommand is similar to the D prefix subcommand. Its primary use is to delete lines from the current line down to, but not including, the line containing the target.

```
DELETE target
```

Examples:

```
DEL /PAPER/
```

```
DEL *
```

20 GET and PUT

A file or part of a file can be included from a disk into the file that is being editing. The GET subcommand inserts data after the current line.

GET fn ft fm firstrec numrec

fn	Filename of the file to be gotten..
ft	Is the filetype. When omitted, the filetype of the file being edited the is assumed.
fm	Is the filemode. When omitted all accessed disks will be searched.
firstrec	The number of the first record to inserted. Default is 1.
numrec	The number of records to be inserted. Defaults to whole file.

Examples:

GET JCL SKEL

GET SAMPLE DIRECT 5 5

The PUT subcommand copies lines from the current line in the file being editing into the one specified on the PUT subcommand.

PUT target fn ft fm

target	Tells XEDIT when to stop writing out lines. Default is 1.
fn,ft	Are as for GET.
fm	Is the filemode. The default is the same as the file being edited.

Examples:

PUT 12 NEW FILE

PUT /EXEC/ OLD FILE B

PUT * = COBOL =

If the file exists, the data PUT is appended to the file. If not, a new one is created.

The PUTD subcommand behaves similarly to the PUT subcommand, except that data is deleted from the file that is being edited as it is being written out.

21 Repeating Commands

There are three XEDIT subcommands available to facilitate the repeating of commands.

<p>& subcommand</p>	<p>When prefixing a subcommand with &, then it will stay in the command area ready to execute again.</p> <p>In order to browse through the file looking for a particular string, this subcommand can be used:</p> <p>&/CMS</p> <p>This will afford the convenience of invoking the ENTER key to reach the next occurrence.</p>
<p>= subcommand</p>	<p>Causes the previous subcommand to be executed again.</p>
<p>? subcommand</p>	<p>Causes the last subcommand to be redisplayed in the command area. The subcommand may then be altered and executed.</p>

22 HELP

If help is required when using XEDIT, the HELP subcommand will connect to the CMS HELP facility.

HELP name

name	The subcommand name.
------	----------------------

CMS HELP context will be displayed and the PF keys will be changed. When exiting HELP, the XEDIT PF keys will be restored.

Alternatively, PF1 is set to display a menu of XEDIT subcommands. It is possible to select which subcommand will require help in the normal way.

23 SET

The SET subcommand is used to change current settings for various XEDIT values.

The keyword SET is optional in these subcommands.

The XEDIT screen has the following format:

PROFILE EXEC A1 V 130 TRUNC=130 SIZE=27 LINE=0 COLUMN=1 message area

prefix
area

```
===== * * * TOP OF FILE * * *      current line (highlighted)
      ....+....1.....+....2.....+....3.....+....4 (scale line) ..... 6.....+....7.
===== &CONTROL OFF
===== & SET UP PF KEYS
===== CP SET PF1 IMM HELP HELP
===== CP SET PF3 IMM #CP DISC      file area
===== CP SET PF9 IMM #CP Q R ALL
===== CP SET PF12 IMM FLIST
===== CP SET PF8 DEPRINT
===== * SET UP PUNCH
===== CP SPOOL PUN VSEIPO
====>      command area
```

X E D I T 1 FILE

file identification area

Certain of these areas (and the format), may be redefined using the following SET subcommands.

The message area is where XEDIT displays error messages. It cannot be changed; however the command line can overlay it. In this case, messages and commands will share the same area.

The prefix area, containing ===== on each line, can be on the left or right of the screen, may be set to contain line numbers, or removed altogether.

```

LEFT
SET PREFIX ON RIGHT

SET PREFIX OFF

SET NUMBER ON
OFF
    
```

The command area is where subcommands are entered, and occupies one and a half lines at the bottom of the screen. The other half of the line contains the status area, which shows the mode you are in. The command area may be moved to the top and combined with the message area or to the bottom, to occupy one line only.

```

ON
SET CMDLINE TOP
BOTTOM
    
```

The file is displayed in the file area. The current line, which is highlighted, is displayed in the middle of the file area.

```

SET CURLINE ON n
    
```

n	Is the required screen line number.
---	-------------------------------------

Below this is the scale line. These lines can be moved anywhere within the file area. The scale line may be removed altogether.

```

SET SCALE ON n
OFF
    
```

n	Is the required screen line number.
---	-------------------------------------

The file identification area is always at the top of the screen. The name, type and mode of the file being edit may be altered using the following SET subcommands:

SET FNAME filetype

SET FTYPE filetype

SETT FMODE filemode

This will effectively create a new file, which must be FILEd or SAVEd.

24 TRUNC and VERIFY

Two other important settings are TRUNC and VERIFY. VERIFY tells the editor which parts of each record you wish displayed.

- C For a COBOL file, columns 1-72 are displayed.
- C For LISTING file, columns 1-121 are displayed.

If a record is longer than the file area on the screen (usually 72 characters), XEDIT will display the record on as many lines as it needs. The SET VERIFY subcommand allows the Verify setting to be altered.

SET VERIFY start end

start	Is the start column to be displayed.
end	Is the last column to be displayed.

Examples:

V 81 133

V 1 72

TRUNC informs XEDIT the last column in which data maybe entered. Most source files, e.g. COBOL, use columns 73-80 for sequence numbers, so the TRUNC setting is column 72.

Any data entered after this column is rejected. Also, data shifted beyond the TRUNC column by inserting data into a line is lost. In order to change the truncation column, use the following subcommand:

SET TRUNC n

n	Is the truncation column. A value of * sets the TRUNC value to the record length.
---	---

Example:

TRUNC 80

25 WRAP and STAY

The WRAP and STAY options affect subcommands that search for targets.

The WRAP function controls where searching for a target is to end.

ON
SET WRAP OFF

ON	Specifies that searching will continue past end-of-file and then continue from the top until the target is found or until the current line is reached.
OFF	Specifies that searching will stop at end-of-file.

The STAY function determines where the current line pointer ends up after an unsuccessful search.

ON
SET STAY OFF

ON	Specifies that the current line pointer remains the same.
OFF	Specifies that the current line pointer goes to end-of-file.

26 Multiple Files

XEDIT allows you to edit more than one file. The only limit to the number of files you can edit is the amount of virtual storage available to you.

The XEDIT subcommand will call another file into storage.

```
XEDIT fn ft fm
```

All files being edited are independent. The operands are as for the CMS XEDIT command. Once the required files have been edited, the XEDIT subcommand can be used for moving from one file to another without operands.

The SET attributes of the files are independent, therefore so screen layouts and other settings may differ for each file. Upon invoking a QUIT or FILE on a file, then XEDIT displays the next file on the screen. When all files have been FILEd or QUITted, XEDIT will return to CMS or FLIST.

The CANCEL macro will cause all files to QUIT. Files that have been changed will not QUIT, therefore QUIT has to be issued in the normal way.

27 Split Screens

The XEDIT physical screen may be split into several logical screens.

This is done by the subcommand:

```
SET SCREEN n
```

n	Number of logical screens that are to be displayed.
---	---

Each logical screen is a replica of a physical screen, except the file area is smaller. If one file is being edited, then the logical screens both contain the file that is being edited. The view of the file may be changed for each screen, in order that all different parts of the same file can be seen. The file can be updated from any logical screen.

When editing more than one file, XEDIT will display as many files as it can.

A common use of this facility is to copy parts of one file to another. XEDIT has the capability for doing this, using temporary files.

In order to write to the temporary file, use the PUT or PUTD subcommands without a field.

Examples:

```
PUT 20
```

```
PUTD /* EOJ/
```

The temporary file can be inserted into another file using the GET subcommand without operands. This technique enables part of one file to be inserted into another.

28 XEDIT and EDGAR

There are differences between EDGAR and XEDIT. However, XEDIT is similar enough to EDGAR to make the transition easy.

When using the EDGAR command on a file, XEDIT will be in EDGAR compatibility mode. By using a special profile macro, XEDIT can be tailored in order that most EDGAR subcommands work in the same way.

In addition, XEDIT subcommands (such as C, M, F, P prefix subcommands) not present in EDGAR may be used.

29 XEDIT and ICCF

XEDIT and ICCF are fairly similar, but you must be aware of the differences, the major ones being:

- C XEDIT reads the file into storage, and changes are only made when writing the file back to the minidisk using the FILE or SAVE subcommands.
- C There are subtle differences between subcommands, such as ICCF's SEARCH and LOCATE. In XEDIT, LOCATE works differently according to the WRAP setting.
- C There is no equivalent of copying lines into a stack in XEDIT. COPY/MOVE subcommands must be blocked or PUT and GET.
- C It may not be possible to edit large files because it is all read into storage first.

30 PF Key Settings

The standard XEDIT program function (PF) key settings are:

PF 1	HELP (XEDIT commands)
PF 2	QUIT
PF 3	TABKEY
PF 5	This PF key is used for selective changes.
PF 6	This PF key is used for selective changes.
PF 7	BACKWARD
PF 8	FORWARD
PF 9	=
PF 10	SPLIT
PF 11	JOIN

The other default settings are rarely used in normal editing.

Chapter 2: XEDIT

Purpose and Function	1
XEDIT: Invoking	2
Standard XEDIT Screen	3
XEDIT: Leaving	4
Editing New Files	5
SAVE	6
Editing: Basic	7
Prefix Subcommands	8
Scrolling Subcommands	9
Targets	10
CHANGE Subcommand	12
Selective Changes	13
COPYING and MOVING	14
BLOCK PREFIX Subcommands	15
FIND	16
SHIFT	17
RIGHT	18
LEFT	19
DELETE	20
GET and PUT	21
Repeating Commands	22
HELP	23
SET	24
TRUNC and VERIFY	27
WRAP and STAY	28
Multiple Files	29
Split Screens	30
XEDIT and EDGAR	31
XEDIT and ICCF	32
PF Key Settings	33