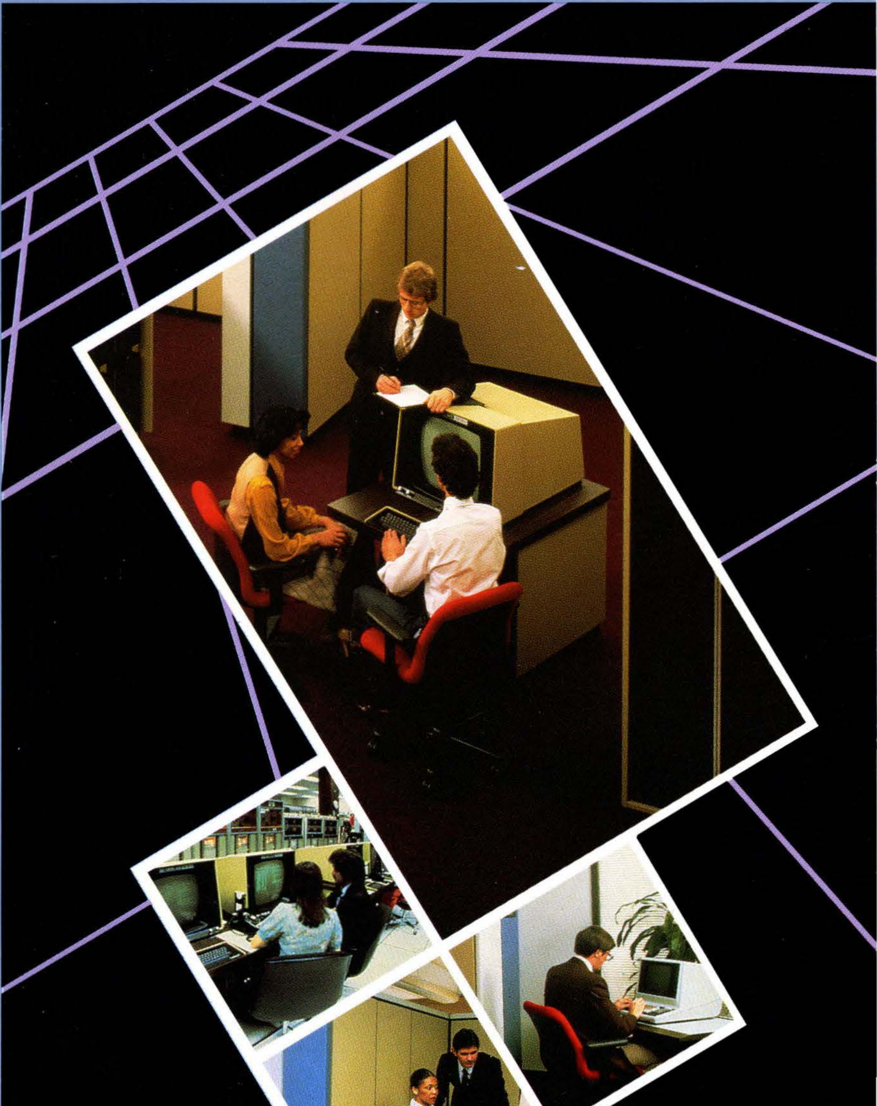
NOS
Full Screen Editor

CONTROL DATA



User's Guide

60460420

Directive Index

Directive	Parameters	Page
ALTER	*END direction string range QUIET*	4-9
BACK	None	4-12
COPY	*range (file 1) TO line (file 2) QUIET*	4-13
DELETE	*range BLANK WORD IN tab QUIET*	4-15
FSE	*file charset GET READ SPLIT*	4-20
GET	*STATUS ALIGN*	4-22
HELP	*directive*	4-25
INSERT	*line PREVIOUS string BLANK WORD*	4-28
LOCATE	*WORD direction string range IN tab UPPER QUIET*	4-30
MOVE	*range (file1) TO line (file2) QUIET*	4-36
PRINT	*range QUIET*	4-38
QUIT	*REPLACE QUIET UNDO PROC /NOS command*	4-40
REPLACE	*WORD direction string1 string2 range IN tab UPPER QUIET*	4-43
SET	Too many parameters to list	4-50
TEACH	Too many parameters to list	4-58
UNDO	None	4-60
VIEW	*line direction SCREEN HOME*	4-61
.	Too many parameters to list	4-63
-	procname *(file)*	4-65
--	comment	4-69
/	NOS command	4-70

Common Parameter Index

The following parameters are common to more than one of the FSE directives. Detailed descriptions of these parameters begin on page 4-5.

Parameter	Values
direction	NEXT num, PREVIOUS num, REPEAT num
(file)	NOS file name of seven or fewer characters.
line	line no., CURRENT, FIRST, LAST, ALL, PREVIOUS num, REPEAT num, NEXT num, line + num, line - num, X, Y, Z, line (file).
range	line no., CURRENT, FIRST, LAST, ALL, PREVIOUS num, REPEAT num, NEXT num, MARK, SCREEN, line + num, line - num, X, Y, Z, line (file), or any two of the preceding parameters.
string	/text/, "text", 'text', \te/xt\, /text1/../text2/

NOS Full Screen Editor

User's Guide

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Publication Number 60460420

Revision Record

Revision B reflects NOS 2.3 released on October 12, 1984 at PSR level 617.

Included in this revision are the additions of new system-defined terminals, reference to the Terminal Definition Utility (TDU), increased explanations of the more advanced functions of FSE (procedure looping and conditional procedures), and an explanation of the use of file marks. There are also numerous technical and editing changes to reflect the enhancements inherent in the new release.

New features and changes, additions, and deletions to information in this manual are indicated by bars in the margins or, if the entire page is affected, by a dot near the page number.

This edition obsoletes all previous editions.

Previous Revision	System Level	Date
A	2.2	October 11, 1983

Preface

The CONTROL DATA® Network Operating System (NOS) provides the capability of editing files page-by-page. The CDC® NOS Full Screen Editor (FSE) enables you to edit sequenced and unsequenced files both page-by-page (screen mode) and line-by-line (line mode).

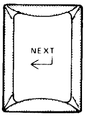
This guide explains how to access and use FSE in both line mode and screen mode. Examples and illustrations help you with each step.

What You Need to Know Before Reading This Guide

This guide is intended for you if you are familiar with the interactive use of NOS. Some knowledge of NOS file types is helpful but is not required. All examples use the CONTROL DATA Viking 721 terminal; knowledge of this terminal is helpful but not essential. For information on the interactive use of NOS, refer to the *NOS Reference Set, Volume 1, Introduction to Interactive Usage*. For detailed hardware descriptions of the Viking 721 terminal, refer to the *721-10/20/30 Hardware Reference Manual* (ordering information is in *Related Publications* later in the preface).

Conventions

Throughout this guide, we use representations of actual keys to show you when you should press a key. For example,



is represented as

NEXT

and

is shown as just

CLEAR

Two keys appearing side by side means hold down the first key while pressing the second. For example,

⇧ CLEAR

means hold down ⇧ and press CLEAR.

Sometimes you have to press more than one key in succession to execute a certain function. This is indicated with a + sign. For example,

⇧ CLEAR + NEXT

means hold down ⇧, press CLEAR, release both, and then press NEXT.

Some function keys have labels both on the keyboard and on the screen. For example, the key labeled only F1 on the keyboard, has the following screen label.

F1 MARK

The screen label is used in this guide.

The keys are sometimes combined. For example, to page backward on the DEC VT100 terminal, you would use:

SHIFT F1 BKW + RETURN

In descriptions of directives or parameters, the shortest valid abbreviations are underlined (usually the first letter). Parameters that must be entered exactly as shown are shown in UPPERCASE CHARACTERS; variable parameters are shown in all lowercase characters. Optional parameters are shown in *italics.*

Anything you enter is shown in blue.

Manual Organization

Section 1 introduces FSE and its capabilities and describes the Viking 721 terminal and the differences between the Viking and other terminals. Through examples, section 2 describes the basic screen editing functions. Section 3 describes the FSE command used to start FSE. Section 4 includes detailed descriptions of the FSE directives. Section 5 shows you a sample session using many of the advanced FSE functions, which are then described in detail in section 6. Using examples, section 7 describes how to use FSE as a line editor. Section 8 contains worksheets to help you figure out how to use FSE on terminals other than the Viking 721.

The order in which you should read these sections depends on the mode of editing you want to use. If you intend to use screen editing, you should read the manual straight through from section 1 through section 7. If you intend to use FSE as a line editor, read section 1 followed by section 3, section 4, and section 7.

Appendix A includes information on the character sets supported by FSE. Appendix B contains explanations of FSE diagnostic messages and how to recover from errors that garble your screen. Refer to appendix C for definitions of unfamiliar terms. Appendix D describes how to use FSE with CDC 722, DEC VT100, Zenith Z19, Heathkit H19, Lear Siegler ADM3A and ADM5, and Tektronix 4115 terminals.† Appendix E is a listing of the directive strings composing the various FSE functions. Appendix F describes the Viking 721 terminal settings needed to ensure proper operation of FSE.

† DEC VT100 is a product of the Digital Equipment Corporation. Zenith Z19 is a product of the Zenith Radio Corporation. Heathkit H19 is a product of the Heath Company. Lear Siegler ADM3A and ADM5 are products of the Lear Siegler Corporation. Tektronix 4115 is a product of the Tektronix Corporation.

Related Publications

The following Control Data publications may help you in further understanding NOS or the Viking 721 terminal. Also listed is a binder to protect your manual. All are available from Control Data sales offices and from Literature Distribution Services, 308 North Dale Street, St. Paul, MN 55103.

Control Data Manual	Publication Number
NOS Version 2 Reference Set, Volume 1, Introduction to Interactive Usage	60459660
NOS Version 2 Reference Set, Volume 2, Guide to System Usage	60459670
NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Screen Formatting Reference Manual	60460430
NOS Version 2 Manual Abstracts	60485500
Software Publications Release History	60481000
721 Display Terminal Operator's Guide/Installation Instructions	62940019
721-10/20/30 Hardware Reference Manual	62940020
721-21/31 Owner's Manual	62950101
Loose Leaf Binder, 6 in by 9 in	60086200

Contents

Introduction 1

Introduction 1

Using the NOS Full Screen Editor (FSE) on a video display terminal, you can display a page of text on the screen, move through a file page by page, and make most of your changes with the touch of a key. To be used as a screen editor, FSE must first be placed in screen mode.

If you are working on a printing terminal, you will use FSE's other editing capability, line editing. In contrast to screen editing, you see only a limited number of lines at any one time. Line editing is also available for use on video display terminals. To be used as a line editor on any type of terminal, FSE must be placed in line mode.

In either mode, you can edit both sequenced and unsequenced files.

FSE Capabilities

Using FSE, you can:

- Create or change a file.
- Undo changes you made to a file during your current terminal session.
- Display and edit two files on one screen.
- Search for and replace text according to the column in which the text appears.
- Move or copy parts of a file within the same file or into another file.
- Search for and replace words.
- Create FSE procedures using FSE directives.
- Manipulate words, lines, and paragraphs of text with the FSE word processing directives.
- Access a tutorial file and practice editing.

These and the many other capabilities of FSE are described in detail in this guide.

Terminals

FSE supports almost all display terminals. First, however, it is necessary that the terminal type be defined to the system. Some terminals already have been defined by Control Data; others are defined by the site or by the user.

Control Data has defined the following terminals.

- CDC Viking 721.
- CDC 722.
- DEC VT100.
- Zenith Z19.
- Heathkit H19.
- Lear Siegler ADM3A.
- Lear Siegler ADM5.
- Tektronix 4115.

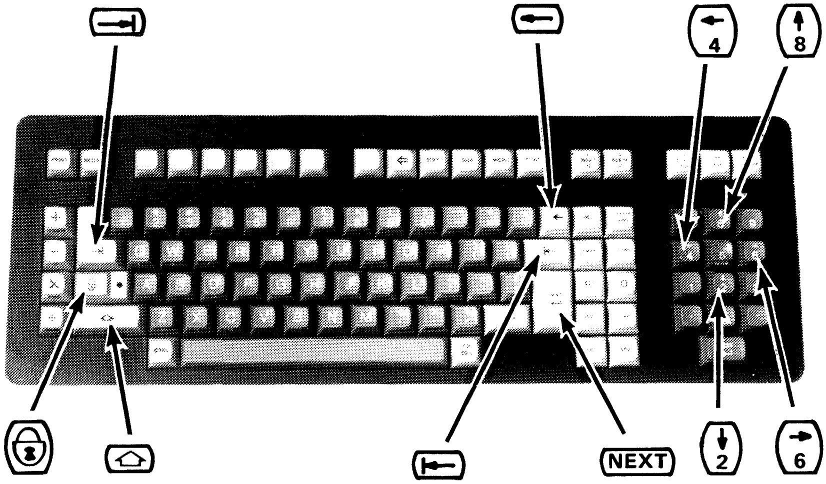
For a terminal other than one of these, check if it has been defined by the site; if not, you may be able to define it yourself. The process is described in the next section under *Getting Started*.

Viking 721 Terminal

One of the terminals already defined to the system is the Viking 721. Throughout this guide, examples and explanations use this terminal as a model for the keys and functions.

To log in to NOS and start FSE, the Viking 721 must be set up correctly. Appendix F shows how the terminal should be set up; however, if your terminal has been used with NOS before, try using it without changing any of the settings.

Some often used keys on the Viking 721 are:



Key	Function
NEXT	Return, carriage return, or new line.
←	Backspace.
⇧	Shift.
🔒	Shift lock.
→|	Tab forward.
|←	Tab backward.
↑ 8 ← 4 → 6 ↓ 2	Cursor movement.

Other Terminals

If you are using a terminal other than the Viking 721, you may have to substitute other keys for those used in the examples. If you have one of the Control Data-defined terminals listed previously, refer to appendix D, *Terminal Support Information*, for a listing of the keys on your terminal that are equivalent to the Viking 721 keys. Section 8, *Using Other Terminals*, provides you with worksheets to record your terminal keys next to the corresponding Viking 721 keys.

The following list shows typical differences between the Viking 721 and other terminals.

Viking 721	Typical Equivalent
NEXT	RETURN or CR
⇧	SHIFT
F1 through F8	PF1 through PF8 or 1 through 8
⇧ F1 through ⇧ F8	Shifted keys on a keypad or no equivalent keys.

The F1 through F8 keys are programmable function keys and are described later. Many terminals require you to press RETURN, NEW LINE, or CR after a programmable function key. For example:

Viking 721	Equivalent on Zenith Z19
F1	F1 + RETURN

Screen Editing 2

Getting Started

To get started on FSE, you must be logged in to NOS. For information on NOS login procedures, refer to the *NOS 2 Reference Set, Volume 2, Guide to System Usage* (ordering information can be found in the preface).

Establishing Screen Mode

After you are logged into NOS, enter the following command to identify your terminal type and tell NOS that you want to be in screen editing mode whenever possible.

```
SCREEN,model
```

NOTE

Screen editing mode remains in effect for the entire terminal session, even if you change editing modes while using FSE.

If you do not enter a SCREEN command prior to starting FSE, FSE will act as a line editor (the default mode). Refer to section 7, *Line Editing*, for information about this function.

model represents the name that identifies your terminal type. If you have one of the following terminals, use the model name shown (Control Data has defined the terminals to the system and assigned the model name).

Terminal	**model**
Viking 721	721
CDC 722	722
DEC VT100	VT100
Zenith Z19 and Heathkit H19	Z19
Lear Siegler ADM3A	ADM3A
Lear Siegler ADM5	ADM5
Tektronix 4115	T4115

For example, to establish screen mode for the Viking 721 terminal, you would enter:

```
SCREEN,721
```

and press:

```
NEXT
```

Other terminals may have been defined to the system by the site. To determine if this is the case, access your site's terminal file library (model is the name of a compiled and stored terminal definition file). Use the commands:

```
GET,TERMLIB/UN=LIBRARY
CATALOG,TERMLIB,R,U,N
```

If your terminal has not been defined, you can do so using the Terminal Definition Utility. Enter the command:

```
GET,TDUIN/UN=LIBRARY
FSE,TDUIN
```

You will get a text file containing pretyped terminal definition statements. Edit the file to define your terminal. You will also need to define your programmable function keys (these keys are described later in this section.) For details, refer to the *NOS Screen Formatting Reference Manual* (ordering information can be found in the preface).

Starting FSE

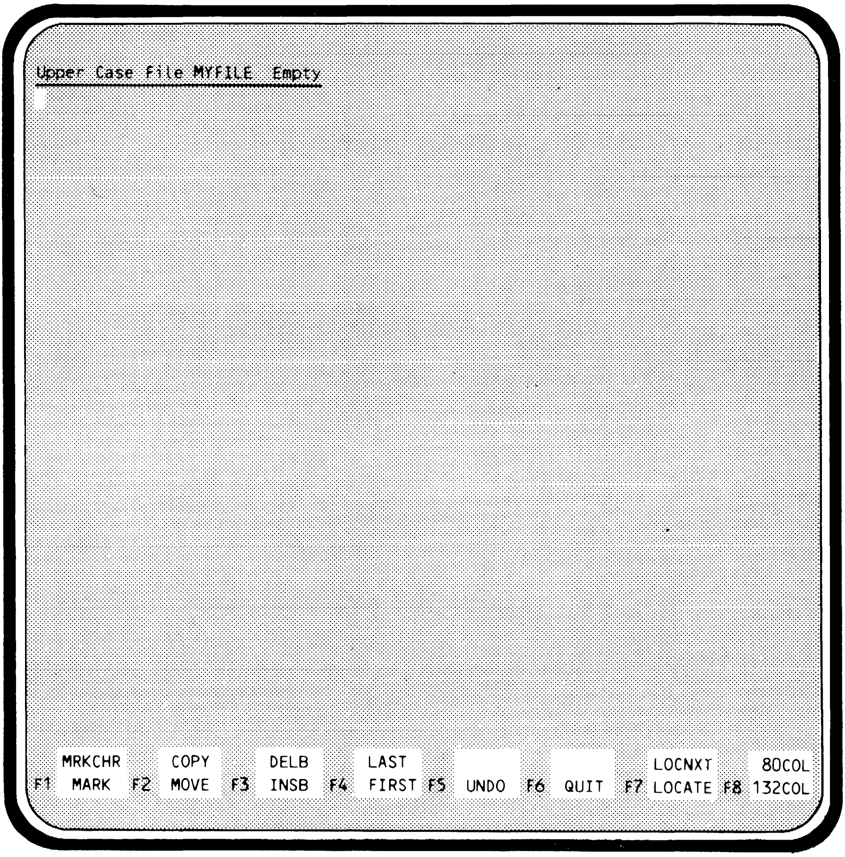
Having established screen mode for your terminal session, you can start FSE and specify the name of the file you want to edit. For example, to start FSE and specify that you want to edit a local file named MYFILE, you would enter:

```
FSE,MYFILE
```

and press:

(NEXT)

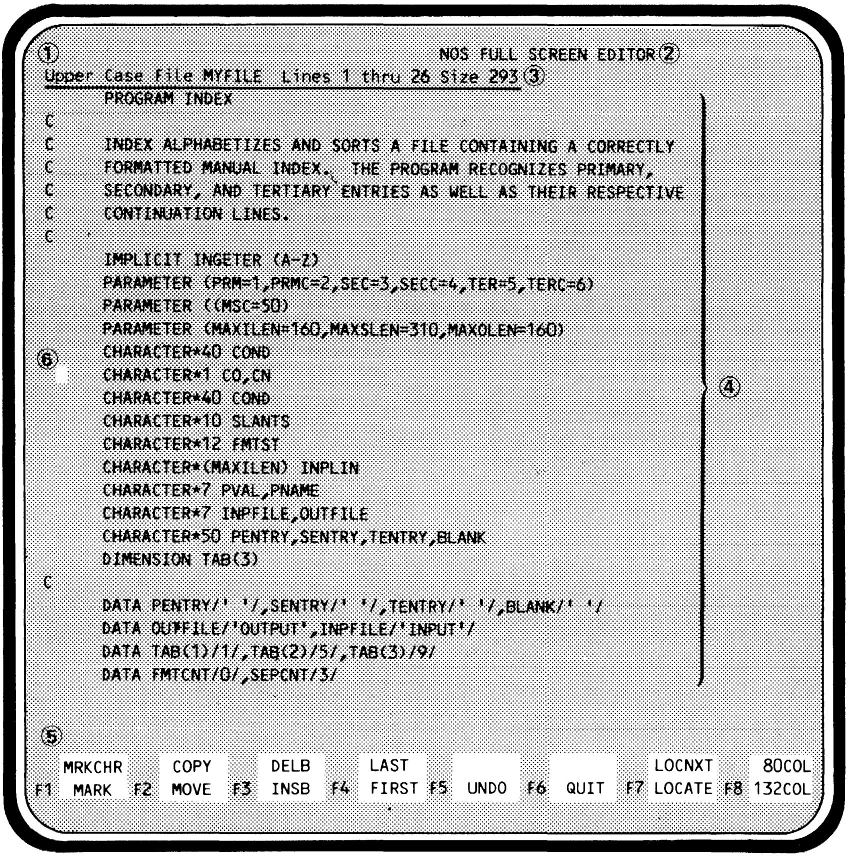
If MYFILE does not already exist as a local file, it is created. Your screen then clears and the following appears.



```
Upper Case File MYFILE  Empty













     MRKCHR       COPY        DELB        LAST                                   LOCNXT       80COL
F1    MARK   F2   MOVE   F3   INSB   F4   FIRST   F5   UNDO   F6   QUIT   F7   LOCATE   F8  132COL
```

Sample Display Screen

The following sample display screen contains a file called MYFILE, which will be used in the demonstrations of editing techniques that follow. It therefore contains intentional errors, which will be corrected later. MYFILE is not released with the system. If you would like to practice as you read, you may want to create this file and enter the appropriate text.



```
                                                NOS FULL SCREEN EDITOR
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INGETER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER ((MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*40 COND
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT',INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/


    MRKCHR      COPY       DELB       LAST                              LOCNXT       80COL
F1  MARK    F2  MOVE   F3  INSB   F4  FIRST  F5  UNDO   F6  QUIT   F7  LOCATE  F8  132COL
```

① Directive Line. The line on which you enter FSE directives. To position the cursor at this line, press HOME or the equivalent key for your terminal.

② Message Area. The area on the right side of the directive line where FSE displays messages and prompts.

③ File Header. The file name, lines currently displayed, and total number of lines in the file. If the file is uppercase, the prefix `Upper Case` appears on this line. If the file is lowercase, no prefix appears.

④ File Text. The contents of the file.

⑤ Programmable Function Key Prompts. The labels currently assigned to function keys F1 through F8. These keys are described later in this section.

⑥ Cursor. Your current position in the file, where text may be entered or edited.

Sample Editing Session

You can do most of your editing by positioning the cursor at the text to be changed and typing the corrections over the old text. Other editing functions can be performed with only the touch of a key. Using basic screen editing techniques, the errors in the previously introduced sample file (MYFILE) will now be corrected. Use this session to obtain an overview of basic screen editing. More advanced aspects of screen editing, including descriptions of the function keys and FSE directives used, appear later in the guide. In the sample session you will:

- Set screen mode.
- Start FSE.
- Position the cursor.
- Type over text.
- Insert and delete characters.
- Insert and delete lines.
- Page forward and backward.
- Move lines.
- Copy lines.
- Undo changes made to a file.
- Access the FSE online HELP file.
- Stop and restart FSE.
- Make changes permanent.
- Edit sequenced files.
- Create multirecord and multifile files.

More detailed information on function keys and FSE directives appears later in this guide.

Setting Screen Mode

After you are logged in to NOS, enter the following command to set screen mode for the Viking 721 terminal.

```
    SCREEN,721
```

Starting FSE

Enter the following command to start FSE and to specify that you want to edit file MYFILE.

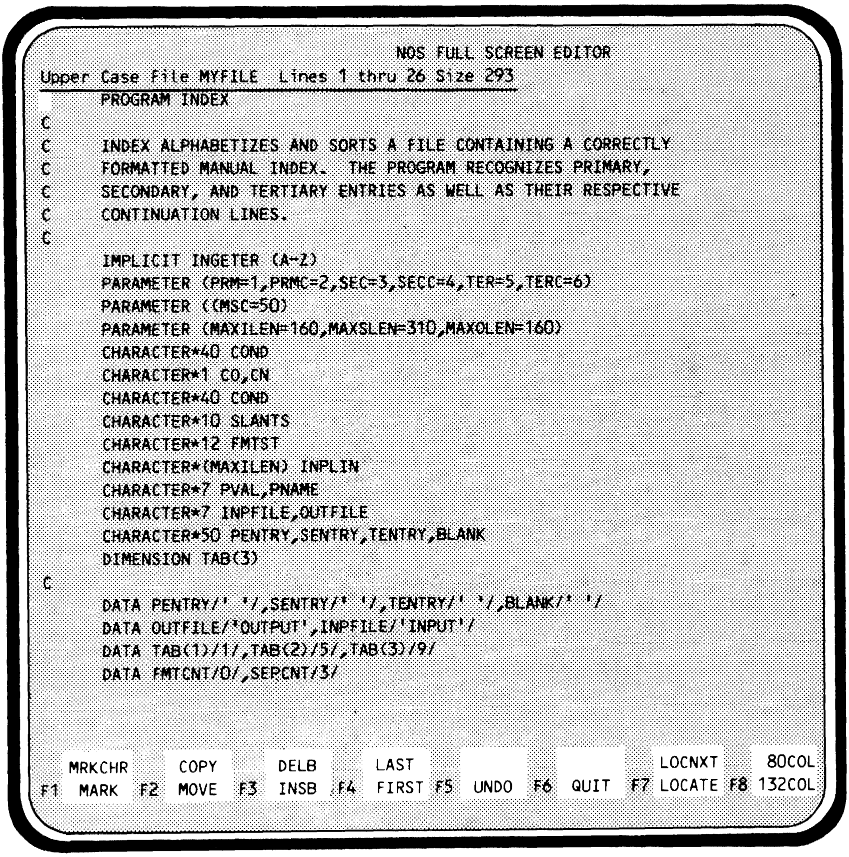
```
FSE,MYFILE
```

FSE accesses file MYFILE. If MYFILE does not already exist as a local file, it is created.

If MYFILE is already a permanent file but is not local, enter the following FSE command to start FSE and make MYFILE local.

```
FSE,MYFILE,G
```

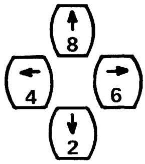
The following screen appears.



```
                                        NOS FULL SCREEN EDITOR
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INGETER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER ((MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*40 COND
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT',INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/


    MRKCHR      COPY      DELB       LAST                              LOCNXT    80COL
F1  MARK    F2  MOVE  F3  INSB  F4   FIRST  F5  UNDO   F6  QUIT   F7   LOCATE F8 132COL
```

Positioning the Cursor

To move the cursor around on the screen, use the arrow keys. On the Viking 721, these keys are located on the numeric keypad to the right of the main keyboard, and appear as:



You do not have to press ⇧ to use these keys and, when you hold them down, they automatically repeat.[†]

Typing Over Text

You can make most corrections to text by positioning the cursor at the text to be corrected and typing the correction over the existing text. For example, in the eighth line of MYFILE, you can change `INGETER` to `INTEGER` by positioning the cursor at the `G`:

```
IMPLICIT INGETER (A-Z)
```

and typing `T`.

```
IMPLICIT INTETER (A-Z)
```

(After you type the `T`, the cursor will move one character forward.)

Then, position the cursor at the second `T`:

```
IMPLICIT INTETER (A-Z)
```

and type `G`.

```
IMPLICIT INTEGER (A-Z)
```

[†] These two characteristics of the terminal are set during terminal installation. If the nonvolatile memory in the terminal fails, they will no longer be available. Information on resetting them can be found in the CDC *721-21/31 Owner's Manual* (ordering information is in the preface).

Inserting Characters

Position the cursor where you want to insert a character and press:

INSRT

For example, the 24th line of file MYFILE is missing a / following 'OUTPUT'.

```
DATA OUTFILE/'OUTPUT',INPFILE/'INPUT'/
```

To add the slash, position the cursor as follows:

```
DATA OUTFILE/'OUTPUT',INPFILE/'INPUT'/
```

Then press:

INSRT

FSE inserts a blank where the cursor is positioned. Type the slash into this blank.

```
DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
```

(After you type the slash, the cursor again moves one character forward.)

Deleting Characters

Position the cursor at the character you want to delete. For example, file MYFILE has an extra parenthesis on the 10th line.

```
PARAMETER ((MSC=50)
```

Position the cursor at the extra parenthesis.

```
PARAMETER ((MSC=50)
```

Then press:

DLETE

The result is:

```
PARAMETER (MSC=50)
```

Inserting Lines

The line:

```
LOGICAL PARMERR
```

needs to be inserted in file MYFILE between the following two lines (lines 20 and 21).

```
CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
DIMENSION TAB(3)
```

To do this, position the cursor where you want the new line to appear.

```
CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
DIMENSION TAB(3)
```

Then press:

⇧ INSRT

FSE inserts a blank line where the cursor is positioned.

```
CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK

DIMENSION TAB(3)
```

You can then type the new line over the blank.

```
CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
LOGICAL PARMERR
DIMENSION TAB(3)
```

Deleting Lines

In file MYFILE, the 12th and 14th lines both declare CHARACTER*40 COND.

```
CHARACTER*40 COND
CHARACTER*1 CO,CN
CHARACTER*40 COND
CHARACTER*10 SLANTS
```

To delete the second of these duplicate lines, position the cursor anywhere on the line to be deleted.

```
CHARACTER*40 COND
CHARACTER*1 CO,CN
CHARACTER*40 COND
CHARACTER*10 SLANTS
```

Then press:

⇧ DLETE

The result is:

```
CHARACTER*40 COND
CHARACTER*1 CO,CN
CHARACTER*10 SLANTS
```

Each time you delete a line, the lines below it move up, leaving a blank line at the bottom of the screen. FSE will fill in these blanks with the appropriate lines from your file when you press:

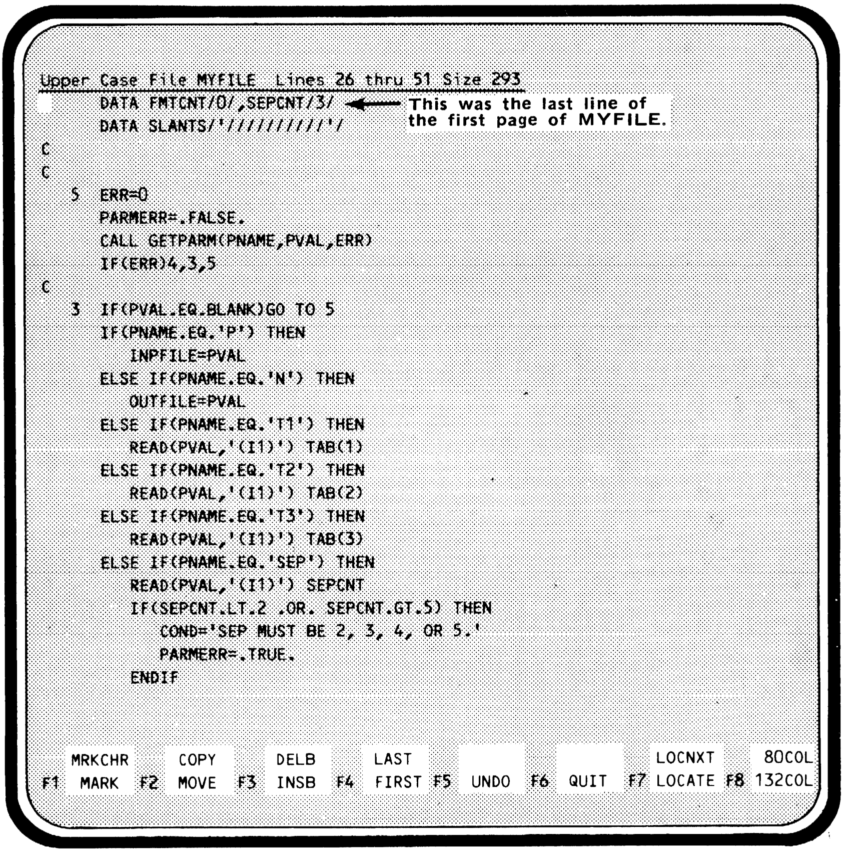
NEXT

Paging Forward and Backward

Pressing:

(FWD)

advances the screen display to the next page forward in the file. The bottom line of the page you were on carries over and reappears at the top of the new display. The following screen is the second page of sample file MYFILE.



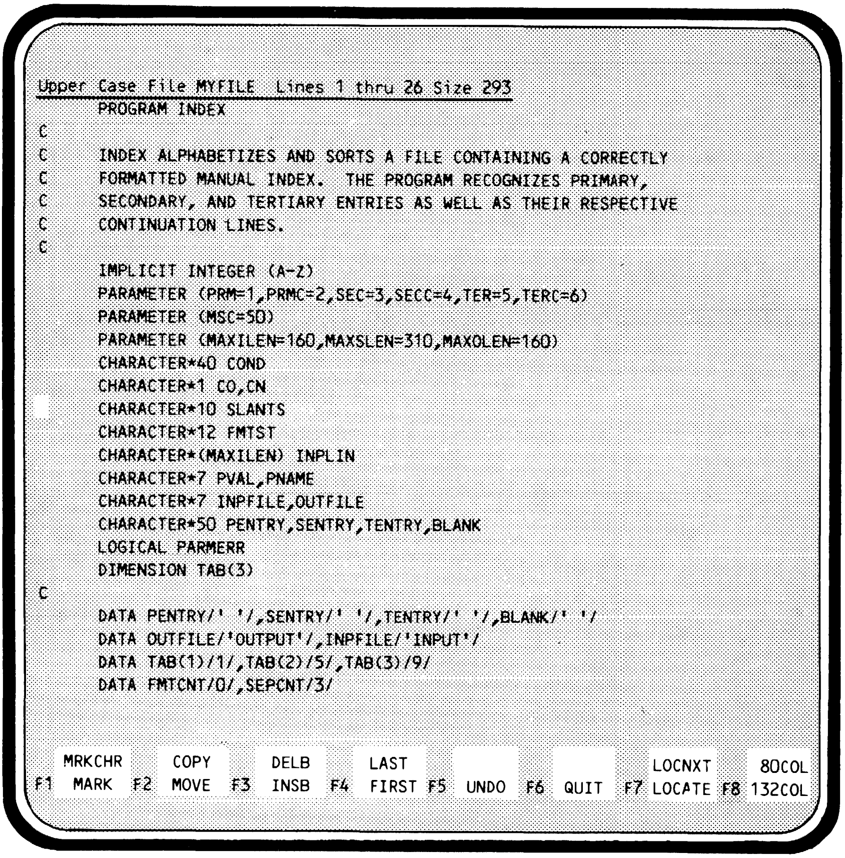
```
Upper Case File MYFILE  Lines 26 thru 51 Size 293
      DATA FMTCNT/0/,SEPCNT/3/
      DATA SLANTS/'//////////'/
C
C
    5 ERR=0
      PARMERR=.FALSE.
      CALL GETPARM(PNAME,PVAL,ERR)
      IF(ERR)4,3,5
C
    3 IF(PVAL.EQ.BLANK)GO TO 5
      IF(PNAME.EQ.'P') THEN
         INPFILE=PVAL
      ELSE IF(PNAME.EQ.'N') THEN
         OUTFILE=PVAL
      ELSE IF(PNAME.EQ.'T1') THEN
         READ(PVAL,'(I1)') TAB(1)
      ELSE IF(PNAME.EQ.'T2') THEN
         READ(PVAL,'(I1)') TAB(2)
      ELSE IF(PNAME.EQ.'T3') THEN
         READ(PVAL,'(I1)') TAB(3)
      ELSE IF(PNAME.EQ.'SEP') THEN
         READ(PVAL,'(I1)') SEPCNT
         IF(SEPCNT.LT.2 .OR. SEPCNT.GT.5) THEN
            COND='SEP MUST BE 2, 3, 4, OR 5.'
            PARMERR=.TRUE.
         ENDIF
```

This was the last line of the first page of MYFILE.

To see the previous page, press:

(BKW)

FSE displays the previous page.



```
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      LOGICAL PARMERR
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/
```

Moving Lines

In sample file MYFILE, the following group of lines needs to be moved.

```
IMPLICIT INTEGER (A-Z)
PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
PARAMETER (MSC=50)
PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
```

To do this, place the cursor at the first line of the group to be moved (line 8).

```
IMPLICIT INTEGER (A-Z)
```

Then press:

```
F1 MARK
```

This marks the line as the beginning of a group of lines to be moved. Then place the cursor at the last line of the group (line 11).

```
PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
```

Press:

```
F1 MARK
```

again. This marks the line as the last of the group to be moved. The entire block marked is highlighted as inverse video display.

```
IMPLICIT INTEGER (A-Z)
PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
PARAMETER (MSC=50)
PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
```

(If you want to move just one line, only the first mark is needed.)

The block is to be relocated between lines 20 and 21.

```
LOGICAL PARMERR
DIMENSION TAB(3)
```

Position the cursor anywhere on the `DIMENSION TAB(3)` line. Then press:

```
F2 MOVE
```

The screen now appears as:

```
LOGICAL PARMERR
IMPLICIT INTEGER (A-Z)
PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
PARAMETER (MSC=50)
PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
DIMENSION TAB(3)
```

The lines you marked no longer exist at their old location, and are no longer highlighted.

Copying Lines

If you want to copy lines without deleting them from their previous location, follow the steps described for moving lines, but press:

```
COPY
```

instead of:

```
F2 MOVE
```

Undoing Changes Made to a File

Suppose at this point, you realize that the lines should not have been moved. Rather than marking the lines again to move them back to their original position, all you need to do is press:

```
F5 UNDO
```

FSE undoes the previous operation. In this example, the lines are moved back to their original position.

```
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
```

F5 UNDO will undo all the changes you made to file MYFILE during the current editing session in the reverse order that they were made. To undo the most recent change, press:

```
F5 UNDO
```

once. To undo the last two changes, press:

```
F5 UNDO
```

twice. You can repeat the step until the original file is restored.

NOTE

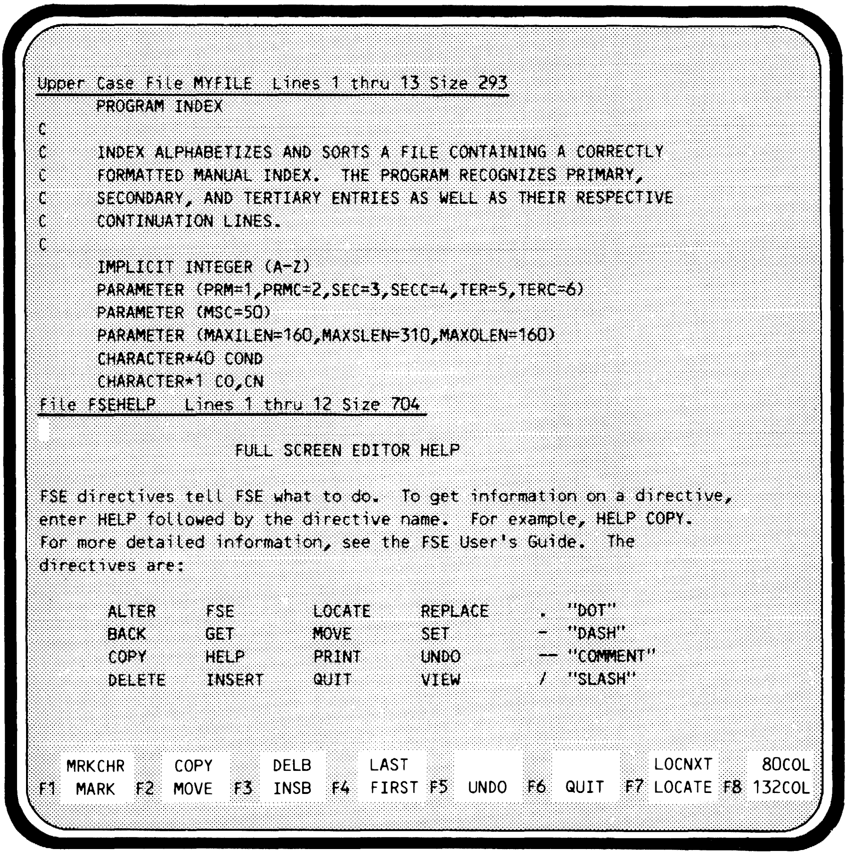
The UNDO command can be turned off using the SET directive (SET UNDO NO). For details, refer to the description of the SET directive in section 4.

Accessing Online HELP

To access online FSE information, press:

HELP

FSE splits the screen in half, with the FSE HELP file on the lower half of the screen.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File FSEHELP  Lines 1 thru 12 Size 704

                      FULL SCREEN EDITOR HELP

FSE directives tell FSE what to do.  To get information on a directive,
enter HELP followed by the directive name.  For example, HELP COPY.
For more detailed information, see the FSE User's Guide.  The
directives are:

        ALTER     FSE       LOCATE    REPLACE     .  "DOT"
        BACK      GET       MOVE      SET         -  "DASH"
        COPY      HELP      PRINT     UNDO        -- "COMMENT"
        DELETE    INSERT    QUIT      VIEW        /  "SLASH"


    MRKCHR    COPY      DELB      LAST                        LOCNXT    80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7  LOCATE F8 132COL
```

By changing the position of the cursor, you can manipulate each half of a split screen separately. For example, to page forward in the HELP file without disturbing MYFILE, move the cursor to the HELP file text and press: FWD.

If you would like to practice, you can edit MYFILE while reading HELP. To return to MYFILE only, press: BACK.

Don't worry if you accidentally make a change to the FSE HELP file; the change is not permanent.

Stopping FSE and Returning to NOS

To return to NOS, press:

```
F6 QUIT
```

Any files you have edited, including changes, remain as local files, ready for use by compilers or other programs. The exception is a direct access file attached in write mode. In this case, any changes are immediately made permanent.

Stopping FSE and Making Your Changes Permanent

To complete your editing session and make any changes to your files permanent, press:

```
HOME
```

This moves the cursor to the FSE directive line. To save the files you've edited as permanent files and to stop FSE, enter:

```
QUIT REPLACE
```

which is abbreviated:

```
QR
```

When you stop FSE, the following messages appear, informing you whether the changes to your file(s) are permanent.

```
FILE:    MYFILE  (PERMANENT)
FILE:    FSEPROC (NO CHANGES)
FILE:    FSEHELP (NO CHANGES) (READ-ONLY FILE)
```

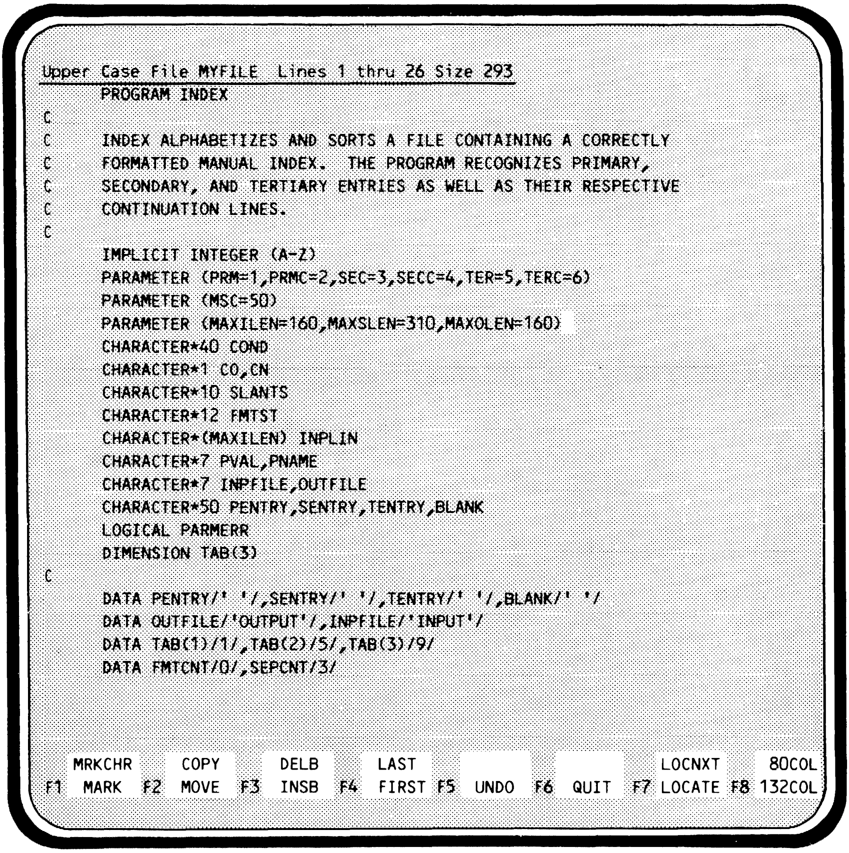
The FSEPROC file is accessed every time you start FSE and is described in detail in section 6, *Advanced FSE Functions*.

Restarting FSE

Having stopped FSE, you can, during your current terminal session, return to editing a file at the exact point you left off. To do this, enter the FSE command without parameters.

```
FSE
```

For example, suppose you finish editing MYFILE at the point shown by the cursor.



```
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      LOGICAL PARMERR
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/



   MRKCHR     COPY     DELB     LAST                      LOCNXT    80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7 LOCATE F8 132COL
```

You then press:

(HOME)

and enter:

QR + (NEXT)

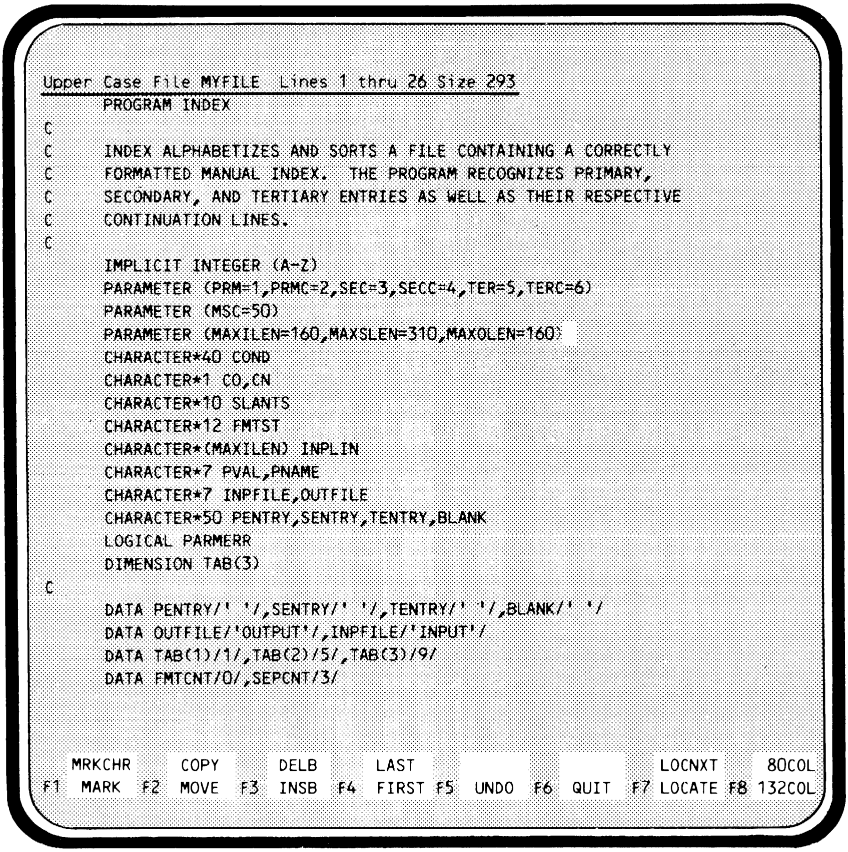
stopping FSE and making your changes to the file permanent. The following messages appear.

```
FILE:  MYFILE  (PERMANENT)
FILE:  FSEPROC (NO CHANGES)
```

After doing other things on NOS, you want to return to editing MYFILE. Enter:

```
FSE + (NEXT)
```

You are returned to the exact point at which you left MYFILE.



```
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      LOGICAL PARMERR
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/


    MRKCHR      COPY      DELB      LAST                         LOCNXT      80COL
F1  MARK    F2  MOVE  F3  INSB  F4  FIRST  F5  UNDO  F6  QUIT  F7  LOCATE  F8  132COL
```

NOTE

If you change your file between FSE editing sessions, be sure to specify the file name when you reenter FSE. For example, enter:

```
FSE,MYFILE
```

rather than

```
FSE
```

Editing Sequenced Files

When you create or edit a sequenced file in the BASIC or FORTRAN subsystem, FSE assumes the file is a numbered sequenced file. Within directives, you can refer to the lines by their sequence numbers, rather than the numbers FSE assigns.

If you change a sequence number by typing over it, FSE does not allow you to make that number less than the preceding nor greater than the following sequence number. If you try to do so, the sequence number is deleted. For example, lines 00140 through 00160 appear as:

```
00140 Line 1
00150 Line 2
00160 Line 3
```

If you type 00165 over 00150, the lines appear as:

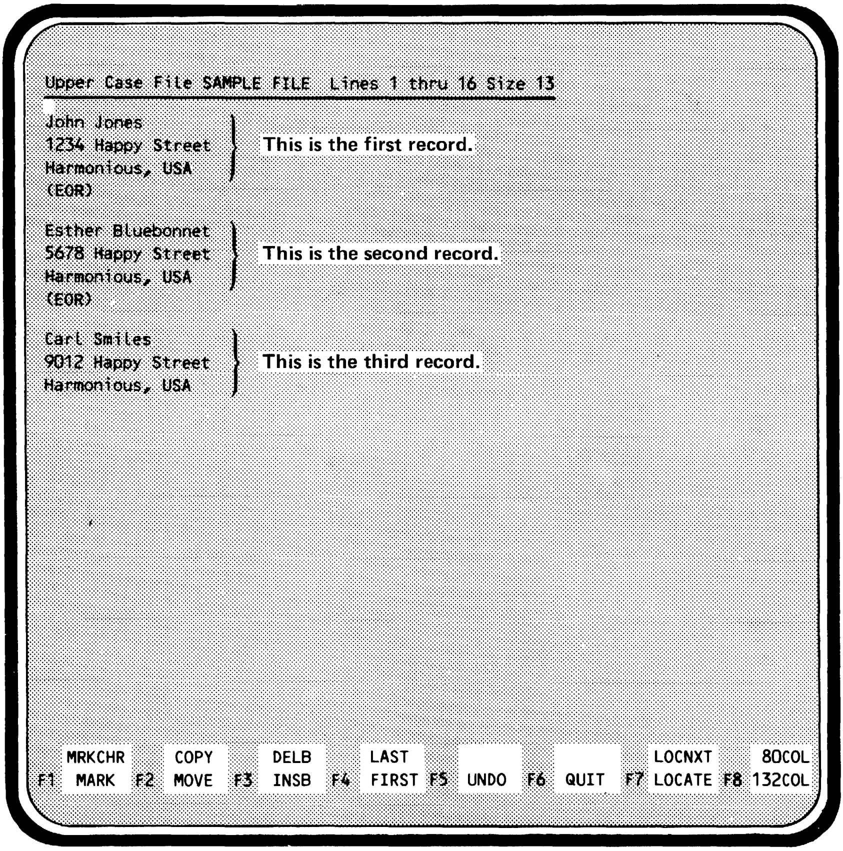
```
00140 Line 1
      Line 2
00160 Line 3
```

The same happens if you try to type a sequence number less than 00140.

Refer to the description of the SET NUMBER directive in section 4 for more information on manipulating sequence numbers.

Creating Multirecord and Multifile Files

If you want to create multirecord or multifile files, you must separate your records and files with marks. End-of-record (EOR) signifies the end of a record; end-of-file (EOF) signifies the end of a file. These marks must be placed on a separate line beginning in column 1. FSE will automatically insert an (EOR) and an (EOF) mark at the end of every edit file that you create (as a result, single-record files don't require file marks, as FSE automatically supplies them). The following is a sample file to demonstrate the placement of file marks in a multirecord file. We will assume you are creating a file containing names and addresses.



```
Upper Case File SAMPLE FILE  Lines 1 thru 16 Size 13

John Jones
1234 Happy Street
Harmonious, USA
(EOR)

Esther Bluebonnet
5678 Happy Street
Harmonious, USA
(EOR)

Carl Smiles
9012 Happy Street
Harmonious, USA
```

This is the first record.

This is the second record.

This is the third record.

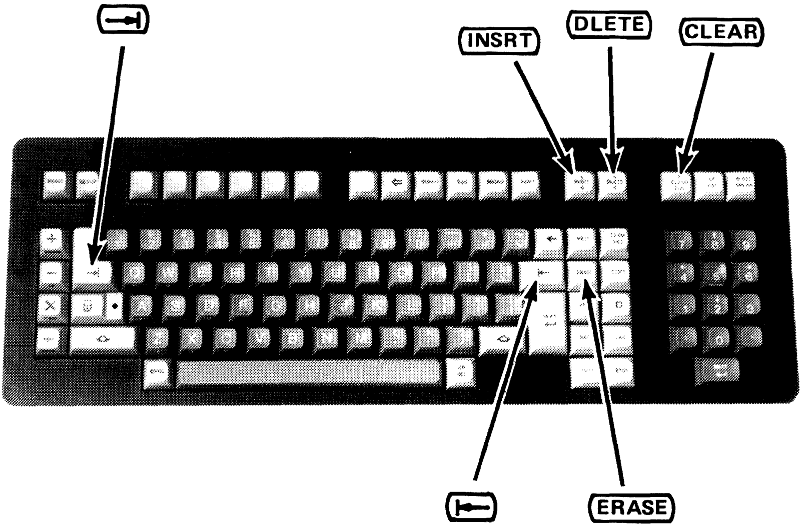
Notice that you don't need an end-of-record mark for the last record in the file. FSE automatically inserts both an end-of-record and an end-of-file mark after the last record.

Viking 721 Terminal Function Keys

In addition to the standard ASCII character keys, like A and 5, FSE supports three types of function keys: editing, CDC standard, and programmable.

Editing Keys

The editing keys are:

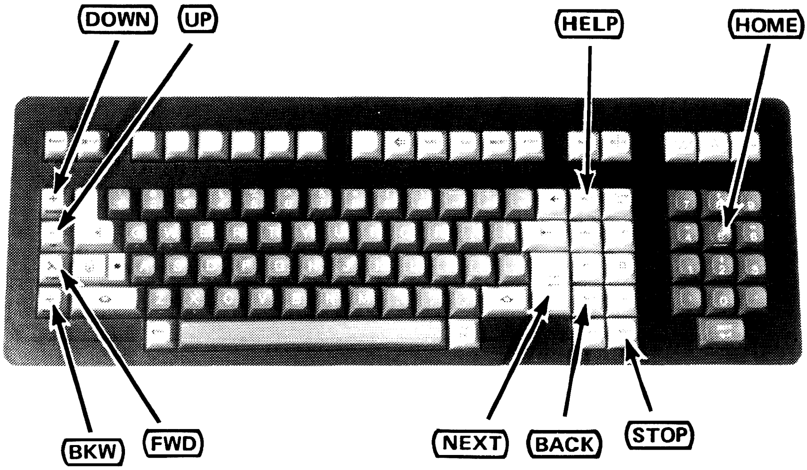


The editing keys usually perform their functions locally (in your terminal) and instantaneously. The editing key functions are:

Key	Function
INSRT	Inserts a blank, allowing you to type in a new character.
⇧ INSRT	Inserts a blank line.
DLETE	Deletes the current character.
⇧ DLETE	Deletes the current line and moves the following lines up. Press: NEXT to fill in the lines at the bottom of the screen.
ERASE	Backspaces a single character and deletes it. When shifted, it deletes the current line and moves the cursor back to the first column.
⇥	Moves the cursor forward to the next tab. Default tab settings are 1, 7, and 72. Refer to the SET directive description in section 4 for information on setting tabs.
⇤	Moves the cursor backward to the immediately preceding tab.
CLEAR	Deletes all characters from the cursor to the end of the line.
⇧ CLEAR	Clears the entire screen; FSE then completely rewrites it. If you suspect the screen does not look right, rewrite it by entering: ⇧ CLEAR + NEXT

CDC Standard Function Keys

The CDC standard function keys are:



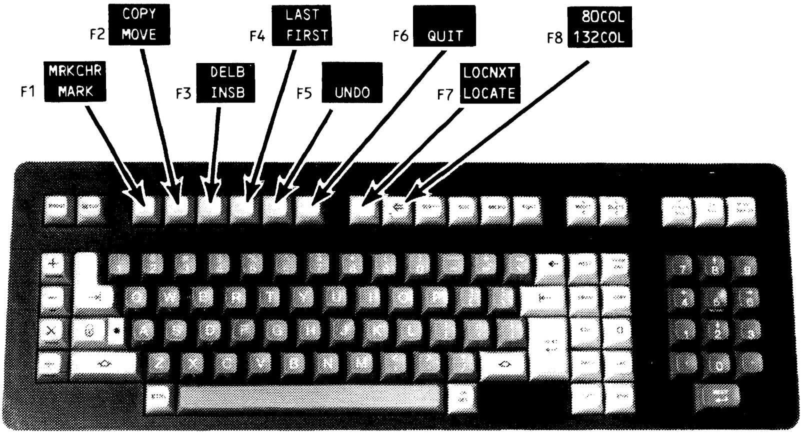
The CDC standard function keys perform operations that apply to nearly all applications. These operations are assigned either to a key or a combination of keys on all supported terminals. The functions are:

Key	Function
FWD	Advances screen one page.
BKW	Moves screen backward one page.
UP	Positions the current line to the top of the screen.
DOWN	Positions the current line to the bottom of the screen.
HELP	Displays the FSE HELP file.
BACK	Returns you to the previously edited file.
STOP	Stops a search or replacement in progress.
NEXT	Terminates an input line.
HOME	Moves cursor to the directive line, enabling you to enter directives.

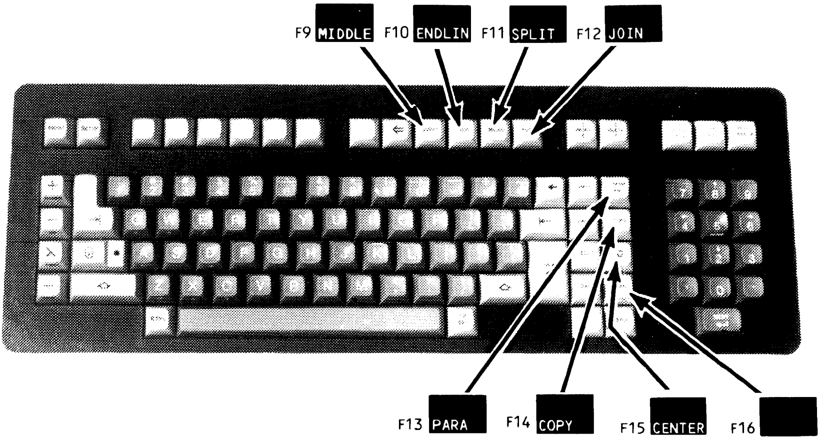
Programmable Function Keys

The programmable function keys can be defined to execute any of the FSE directives (refer to section 6, *Advanced FSE Functions*). The default functions of the programmable function keys are displayed at the bottom of your screen.

Usually, only the F1 through F8 function key prompts are displayed. The default prompts and their corresponding keys are:



The F9 through F16 prompts can be displayed using the SET PROMPT directive described in section 4. The default prompts and their corresponding keys are:



The lower line of the prompt indicates the unshifted key function; the upper line indicates the shifted key function. The function key numbers are in raised letters adjacent to the keys. Some of the programmable function keys also have labels on the key itself. Their labels are used by other applications and have no significance within FSE. The default functions for the Viking 721 programmable function keys are:

Key **Description**

MRKCHR
F1 MARK

Pressed once, marks a line for future use with another function. Pressed twice, marks a range of lines. If you accidentally mark the wrong line or line range, press:

F5 UNDO

to unmark the line or range. When shifted,

F1 MRKCHR

marks a range of characters.

COPY
F2 MOVE

Moves the range of lines set by the MARK function, or characters set by the MRKCHR function, to the position marked by the cursor. When shifted,

F2 COPY

copies the range of lines or characters set by the MARK or MRKCHR functions.

DELB
F3 INSB

Inserts 22 blank lines at the position marked by the cursor. When shifted,

F3 DELB

deletes blank lines, starting with the line the cursor is at, until a nonblank line is encountered.

LAST
F4 FIRST

Positions the cursor at the first line in the file. When shifted,

F4 LAST

positions the cursor to the last line in the file.

Key	Description
F5 UNDO	Cancels any marks you have set. If you set no marks, F5 UNDO undoes the most recent change made to your file during your current editing session. Pressing: F5 UNDO again removes the next most recent change, and so on.
F6 QUIT	Stops the current editing session. If your file is an indirect access file, the changes you've made will not become permanent when you end the terminal session. If your file is direct access, the changes will become permanent.
F7 LOCNXT LOCATE	Prompts you to enter the text you want to locate. When you enter the text and press: (NEXT) FSE locates the text, positioning the cursor at the first character of the text string. When shifted, F7 LOCNXT locates the next occurrence of the last text located.
F8 80COL 132COL	Sets the terminal to 132-column mode. When shifted, F8 80COL sets the terminal to 80-column mode (default).
F9 MIDDLE	Positions the line the cursor is at to the middle of the screen.
F10 ENDLIN	Moves the cursor to the end of the line it is on.
F11 SPLIT	Splits the line the cursor is on into two lines. If the cursor is at the beginning of the line, it inserts a blank line above it. Otherwise, it splits the line at the cursor position.
F12 JOIN	Joins the line the cursor is at with the next line.

Key	Description
F13 PARA	Reformats lines you marked to conform to margins set with the SET WORD FILL directive (default margins are 1 and 65). If you set no marks, F13 PARA reformats the paragraph the cursor is at. Paragraphs are delimited by blank lines. Refer to the description of the SET WORD FILL directive in section 4 for more information.
F14 COPY	Copies the lines or characters set by the MARK or MRKCHR function.
F15 CENTER	Centers the line the cursor is on according to the boundaries set by the SET WORD FILL directive (described in section 4). Default margins are 1 and 65.
F16	Undefined.

For details on how to define or redefine the programmable function keys, refer to section 6, *Advanced FSE Functions*.

For information on default programmable function key settings for other supported terminals, refer to appendix D, *Terminal Support Information*.

Entering Directives

There will be occasions when the functions provided by the previously described terminal keys do not meet your editing needs. In addition to function keys, FSE provides a number of directives (the directives are described in detail in section 4, *FSE Directives*).

To enter a directive, press:

HOME

This positions the cursor on the FSE directive line at the top of the screen; there you enter the FSE directive followed by:

NEXT

The directive uses the position the cursor was in just before you pressed:

HOME

If FSE does not recognize your entry as a valid FSE directive, it positions the cursor at the point where the directive becomes unrecognizable. For example, suppose you entered a misspelled form of the UNDO directive.

```
UMDO
```

FSE positions the cursor at the M because that is the point at which it could not recognize the entry.

```
UMDO
```

You then type the correction over the mistake and press:

NEXT

to try again. If you press:

NEXT

without making a correction, FSE still does not recognize the directive, and will ignore it, erasing the directive line. Check the spelling of the directive and try again.

FSE Command 3

The FSE command starts the Full Screen Editor. With this command you tell NOS which file is to be edited and other, optional, information. The optional parameters are shown in italics in the following descriptions; the shortest valid abbreviations for the values are underlined.

You can use two formats for the FSE command. The first format is order-independent, that is, you can enter parameters in any order. The format is:

> FSE,*FN=filename,OP=access,I=input,L=output,IP=procedure,*
> *WF=workfile.directive(s)*

The second is order-dependent, that is, you must enter the parameters in the order shown. The format is:

> FSE,*filename,access,input,output,procedure,workfile.directive(s)*

Parameter	Description
FN=filename	Specifies the file you want to edit. filename must be a local file, unless you specify the OP=GET parameter. If you do not include a file name on the FSE command, FSE either resumes your previous editing session or, if you had no previous editing session, prompts you for a file name.
OP=access	Specifies the character set and/or the location of the file to be edited. The following are valid access parameter entries.

Value	Description
DISPLAY	Specifies 6-bit display code.
NORMAL	Specifies 6-bit display code (default if your terminal is in normal mode).
ASCII	Specifies NOS 6/12-bit display code (default if your terminal is in ASCII mode).
ASCII8	Specifies 7-bit ASCII code right-justified in a 12-bit byte.
GET	Makes the specified file local by either getting an indirect access file or attaching a direct access file.

Parameter	Description
	Both character set and location can be specified as single-letter abbreviations, and must not be separated by commas, for example: `OP=GA  or GA` When you specify access, you must also specify a file name.
I=input	Specifies an input (directive) file other than the default file INPUT.
L=output	Specifies an output (listing) file other than the default file OUTPUT.
IP=procedure	Enables you to change the default procedure library from FSEPROC to the file of your choice.
WF=workfile	Enables you to specify a work file other than the default file ZZZWORK.
directive(s)	Allows you to enter FSE directives within the FSE command itself. This parameter must be preceded by a period. To enter more than one directive on the FSE command, separate them with a semicolon. The directives can use all of the command line; if you have sequence numbers on the right end of command lines, terminate the last directive on the line with the delimiters ;--.

If neither the input file nor output file you specify is assigned to your terminal, FSE executes in batch mode.

NOTE

When FSE executes in batch mode, the input file must be in the 6/12-bit display code format. To increase the efficiency of batch processing, set the undo feature to NO. (For details, refer to the description of the UNDO parameter of the SET directive in section 4).

Before entering the FSE command, you determine the editing mode by entering either the NOS SCREEN or the NOS LINE command. During an editing session, you can change to the other mode of editing by using the SET directive (described in section 4). This directive affects only the editing session during which it is entered. The NOS command you entered before starting FSE resumes effect for the following editing session. This only applies if you resume a session. If you start a session, you will return to line mode (the default).

For example, suppose you enter the NOS SCREEN command and start FSE, editing file MYFILE. Then, within FSE you use the SET LINE directive to switch to line editing. After line editing for a while, you enter the QUIT directive. The next time you enter the FSE directive without parameters (to resume your FSE editing session), you are returned to file MYFILE in screen mode (your original NOS SCREEN command).

FSE Directives 4

FSE directives tell FSE what to do. When screen editing, you enter directives on the directive line. When line editing, you enter directives after the ?? prompt. In both modes, you must press:

(NEXT)

after the directive to begin execution.

Directive Syntax

Most directives begin with a verb, followed by qualifiers called directive parameters. (Some directives begin with a special character followed by a verb.)

You can often figure out how to specify a directive by thinking of it as a sentence. For example:

Sentence	Directive
Locate all occurrences of *abc*.	`LOCATE ALL /abc/`
Move line 12 to just after line 50.	`MOVE 12 TO 50`
Replace the next three occurrences of *abc* with *xyz*.	`REPLACE NEXT 3 /abc/xyz/`

Abbreviations and Spaces

In directives, you can abbreviate words (other than file names) by specifying either the first letter or the first three letters. An entry can use uppercase letters, lowercase letters, or a combination. For example, you can enter LOCATE as:

```
L
Loc
Locate
```

Although spaces or commas are required between numbers, they are not required between words. Using abbreviations and eliminating unnecessary spaces results in compact entries, as in the following examples.

Long Version	Compact Version
`COPY 20 TO 50`	`C20T50`
`MOVE 20 50 TO 100`	`M20 50T100`
`LOCATE ALL WORD/XVAR/`	`LAW/XVAR/`

Parameter Order

With few exceptions, you can specify directive parameters in any order. For example, the following directives all locate each occurrence of the word XVAR.

```
LAW/XVAR/

L/XVAR/AW

LW/XVAR/A
```

String Delimiters

In the preceding directives, slashes serve as delimiters for the string XVAR. However, if a string is the last parameter on a directive line, and the string does not end in a blank, you can omit the closing string delimiter, as, for example, in the directive:

```
LAW/XVAR
```

NOTE

In batch jobs and FSE procedures, do not omit the closing string delimiter.

Misspelled Directives

FSE does not recognize that a directive is misspelled. Instead, it assumes that any directive it does not recognize is abbreviated and reads as parameters any characters following the first. For example, suppose you enter the following misspelled COPY directive.

```
CORY 10 TO 50
```

Because FSE does not recognize a directive, it assumes you used the C abbreviation for COPY and then tries to read the O as a parameter. In this case, there is no COPY parameter that can be abbreviated to O, so FSE displays an error message.

Combining Directives

By separating directives with semicolons, you can enter more than one directive on a line. The following example is a combination of the LOCATE and VIEW directives.

```
L/SUBROUTINE/;V
```

If there is an error in the syntax of the first directive, FSE stops and displays an error message; it does not execute the second directive.

Conventions

When describing the FSE directives and parameters in the remainder of this section, we use the following conventions.

- The shortest allowable abbreviated entry is underlined.
- Optional parameters are in *italics*. In most cases, it appears that all the parameters are optional. This is technically true, but once you include a particular parameter, others may no longer be optional.
- Examples show the directives and parameters abbreviated in uppercase letters.
- To make the examples more readable, spaces are left between many parameters, although they are not required. For example, the directive:

  ```
  MOVE 20 50 TO 100
  ```

  could be entered as:

  ```
  M20 50T100
  ```

  but would appear in examples as:

  ```
  M 20 50 T 100
  ```

Common Parameters

Most directives use at least one parameter from the following common set. Using these parameters with a particular directive is described later in this section. The detailed descriptions of the possible values for each parameter follow.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>direction</td><td>Specifies the direction FSE is to move through the file and the number of times it is to execute the directive. The default is forward. Possible values and their descriptions are:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>NEXT num</td><td>Instructs FSE to move forward through the file, beginning the specified operation on the next line and executing it num times. The default setting for num is 1.</td></tr>
<tr><td>PREVIOUS num</td><td>Instructs FSE to move backward through the file, beginning the specified operation at the previous line and executing it num times. The default setting for num is 1.</td></tr>
<tr><td>REPEAT num</td><td>Specifies the number of times an operation is to be executed starting at the current line. REPEAT always moves forward. The default setting for num is 1.</td></tr>
</table></td></tr>
<tr><td>(file)</td><td>Specifies a NOS local file name. NOS file names can be seven or fewer alphanumeric characters. The default is the current file. The parentheses are required.</td></tr>
</table>

Parameter	Description
line	Specifies the line or lines to be affected by the directive. The default line setting is the current line. Possible values are:

Value	Meaning
line number	The number of a particular line within the file. In an unsequenced file, line number 1 is the first line of the file, line number 10 is the tenth, and so on. In a sequenced file, line number is the sequence number of the line, for example, line number 20 is the line assigned the sequence number 20, rather than the 20th line of the file (they may be the same, but not necessarily).
ALL	All lines in the file.
direction	One of the direction parameter values: NEXT num PREVIOUS num REPEAT num num is the number of times the directive is to be executed.
CURRENT	The current line, that is, either the line the cursor is on (for screen editing) or the last line displayed (for line editing).

Value	Meaning
FIRST	The first line in the file.
LAST	The last line in the file.
line + num	The number of a particular line plus a specified value. For example, in an unsequenced file, 38 + 45 specifies the 83rd line in the file; in a sequenced file, 38 + 45 specifies the 45th line after the line with the sequence number 38 (this may not be the 83rd line). For both file types, C + 10 specifies the 10th line after the current line.
line - num	The number of a particular line minus a specified value. For example, in an unsequenced file 83 - 45 specifies the 38th line in the file; in a sequenced file, 83 - 45 specifies the 45th line before the line with the sequence number 83 (this may not be the 38th line). For both file types, L - 1 specifies the next to last line in the file.
X or Y or Z	Specifies the X, Y, or Z position pointer. Refer to the SET directive for information on the X, Y, and Z pointers.
line (file)	Specifies a line from another file. Any of the line parameter values can be followed by a file name. The parentheses are required.

Parameter	Description
range	Specifies either a line or a range of lines to be affected by the directive. The default is the current line. Valid range parameter entries are:

Value	Meaning
line	One of the line parameter values.
line line	A group of lines delimited by two line parameter values.
MARK	Lines marked with the F1 MARK key or the SET MARK directive.
SCREEN	Lines appearing on the current screen (screen mode only).

Parameter	Description
string	Specifies a text string. The default string parameter is the last specified string. If a string parameter is the last parameter on a directive, you do not need a closing delimiter. Valid string parameter entries are:

Value	Meaning
/text/	Specifies the string text.
"text"	Specifies the string text.
'text'	Specifies the string text.
\te/xt\	Specifies the string te/xt.
/text1/../text2/	Specifies the string beginning with text1 and ending with text2 on the same line and including any text between.

In the remainder of this section we list all the FSE directives in alphabetical order, show their formats, and describe their parameters. For a parameter that is one of the common set, refer to either the *Common Parameter Index* on the inside front cover for a listing of the possible entries or to *Common Parameters* earlier in this section for detailed descriptions.

ALTER

Enables you to change lines by entering either a string or one of the modification characters (described later).

Without Parameters

When you are screen editing and you enter ALTER with no parameters, FSE prompts you with:

```
<-- ALTER WHAT?
```

You then enter the text or modification character to change the current line. The modification characters are:

Modification Character	Effect
#	Deletes a character.
&	Replaces a character with a space.
!	Erases the current character and any following characters. Anything entered after ! is moved to the end of the shortened line.
^characters#	Inserts the string characters before the character the ^ is positioned under.

When you enter an alphanumeric character other than those listed here, it replaces the existing character.

When you are line editing and you enter ALTER with no parameters, FSE displays the current line and prompts you for changes, as for example:

```
 75  DIMENSIOM TAB(33))xx
A??
```

You then space over to the position under the character to be edited and use one of the modification characters to make your change.

In this example, you could make the following changes.

```
 75  DIMENSIOM TAB(33))xx
A??          N      )!
```

The line appears as:

```
75  DIMENSION TAB(3)
```

Format

ALTER *END direction string range QUIET*

Parameters

END

Adds a string at the end of a line. For example,

```
A E/abc/
```

adds the string abc to the end of the current line.

direction

Specifies the direction ALTER is to move and the number of times the directive is to be executed. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

The following ALTER directive:

```
A P 2/#/
```

deletes the first character of the two previous lines.

string

Specifies the text or modification characters to change or replace the current line. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

A ^ at the end of an ALTER string tells FSE to change the line and then prompt you for more changes.

range

Specifies the lines to be altered. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

The following ALTER directive:

```
A A/^ #/
```

inserts a space at the beginning of all lines in the file.

If you specify a group of lines on an ALTER directive, you are prompted only once; the same corrections are made to all the lines in the group.

A primary use of ALTER during screen editing is to indent blocks of text. For example, the following ALTER directives perform the described function on lines you have previously marked.

Directive	**Function**
`A M/###/`	Deletes the three leftmost characters from all marked lines.
`A M/^   #/`	Inserts three spaces at the beginning of the marked lines.

QUIET

Instructs FSE not to display the results of the ALTER directive.

BACK

Returns you to a file you already edited during the current editing session.

Format

BACK

Example

You are line editing file MYFILE in line mode and enter `HELP`. This accesses the FSE HELP file and prints the first nine lines.

```
?? h
   1
   2                      FULL SCREEN EDITOR HELP
   3
   4 FSE directives tell FSE what to do.  To get information on a directive,
   5 enter HELP followed by the directive name.  For example, HELP COPY.
   6 For more detailed information, see the FSE User's Guide.  The
   7 directives are:
   8
   9       ALTER     FSE        LOCATE     REPLACE     .  "DOT"
??
```

To return to editing file MYFILE only, either enter the BACK directive or press:

BACK

on the Viking 721 terminal. FSE remembers the last seven files you edited. By successively entering:

```
BACK
```

or pressing:

BACK

you can return to all seven files. The directive will not return you to a file if you have ended the editing session. Refer to the FSE directive description for another way to access a previously edited file.

COPY

Copies lines from one location to another, either within a file or between two files.

Without Parameters

If you do not specify parameters, the copied line appears immediately below the original.

Format

COPY *range (file1) TO line (file2) QUIET*

Parameters

range

Specifies the lines to be copied. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

(file1)

Specifies the file containing the lines to be copied. Any NOS file name is valid. The default is the current file.

TO

Separates the copied lines from their destination.

line

Specifies the line after which the copied lines will appear. This can be either a line in the current file or, when combined with the (file2) parameter, a line in another file. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

For example, to copy lines 17 through 25 immediately after line 65, enter:

```
C 17 25 TO 65
```

When you are line editing, FSE displays the copied lines with their new line numbers.

```
66   6   IF(.NOT.PARMERR) GO TO 5
67       OPEN(3,FILE='OUTPUT')
68       WRITE(3,*) COND
69       STOP 'INVALID PARAMETER.'
70 C
71   4   IF (TAB(1).GT.TAB(2) .OR. TAB(2).GT.TAB(3)) THEN
72           COND='INVALID TABS (T1>T2 OR T2>T3).
73           GO TO 6
74       ENDIF
```

When you are screen editing, FSE moves the cursor to the first copied line in its new position.

When you are screen editing and use the COPY key, it copies before, rather than after, the line indicated, because COPY executes the following directive.

`C M TO P` (COPY MARK TO PREVIOUS)

(file2)

Specifies the file (any NOS file name) to which the lines are to be copied; used only if the destination is other than (file1). For example, the following directive copies all of file X to the end of file Y.

```
C A (X) TO L (Y)
```

To specify the line after which the copied lines are to appear, include the line parameter before the (file2) parameter. For example, to copy lines 10 through 50 of file MYFILE (the current file) to after line 20 of file Y, enter:

```
C 10 50 TO 20 (Y)
```

QUIET

Instructs FSE not to display the results of the COPY directive.

DELETE

Deletes one or more lines.

Without Parameters

If you do not specify any parameters, the current line is deleted.

Format

DELETE *range BLANK WORD IN tab QUIET*

Parameters

range

Specifies the lines to be affected by the directive. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

For example, to delete lines 30 through 33, enter:

```
D 30 33
```

In line editing mode, lines 30 through 33 are then printed, showing you the lines you have deleted.

```
30     5  ERR=0
31        PARMERR=.FALSE.
32        CALL GETPARM(PNAME,PVAL,ERR)
33        IF(ERR)4,3,5
```

When you are screen editing, the lines just disappear.

After a DELETE directive, the current line is the line just before the deleted lines. In the preceding example, that would be line 29.

When you are screen editing, a quick way to delete a group of lines is to mark the first and last lines of the group using:

```
F1 MARK
```

Then enter:

```
D M
```

and the marked lines are deleted.

DELETE followed by INSERT can be used to replace lines. For example,

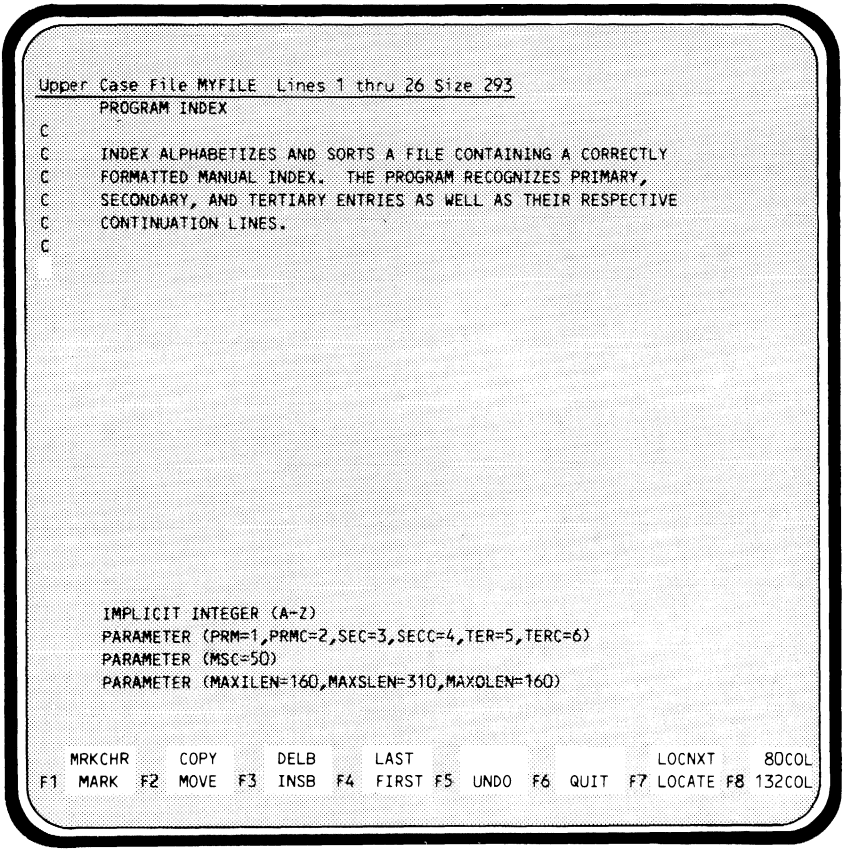
```
D;I
```

deletes the current line and prompts you to insert new lines in its place.

BLANK

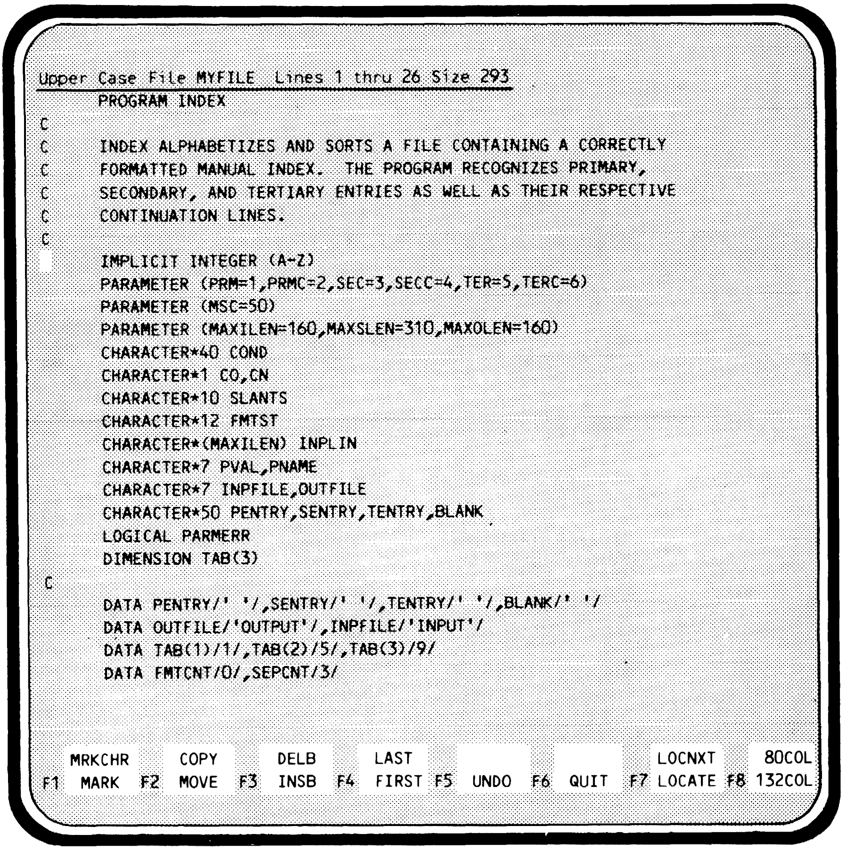
Deletes all blank lines, from the current cursor position to the first nonblank line.

In the following example, to delete all the blank lines but one, you would position the cursor on the second blank line and enter `DELETE BLANK` (or press `F3` **DELB**).



```
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C















      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)

   MRKCHR     COPY      DELB      LAST                               LOCNXT      80COL
F1 MARK   F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7 LOCATE F8 132COL
```

The following screen results:



```
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      LOGICAL PARMERR
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/



   MRKCHR      COPY      DELB      LAST                             LOCNXT      80COL
F1  MARK   F2  MOVE  F3  INSB  F4  FIRST  F5  UNDO  F6  QUIT  F7  LOCATE  F8  132COL
```

WORD

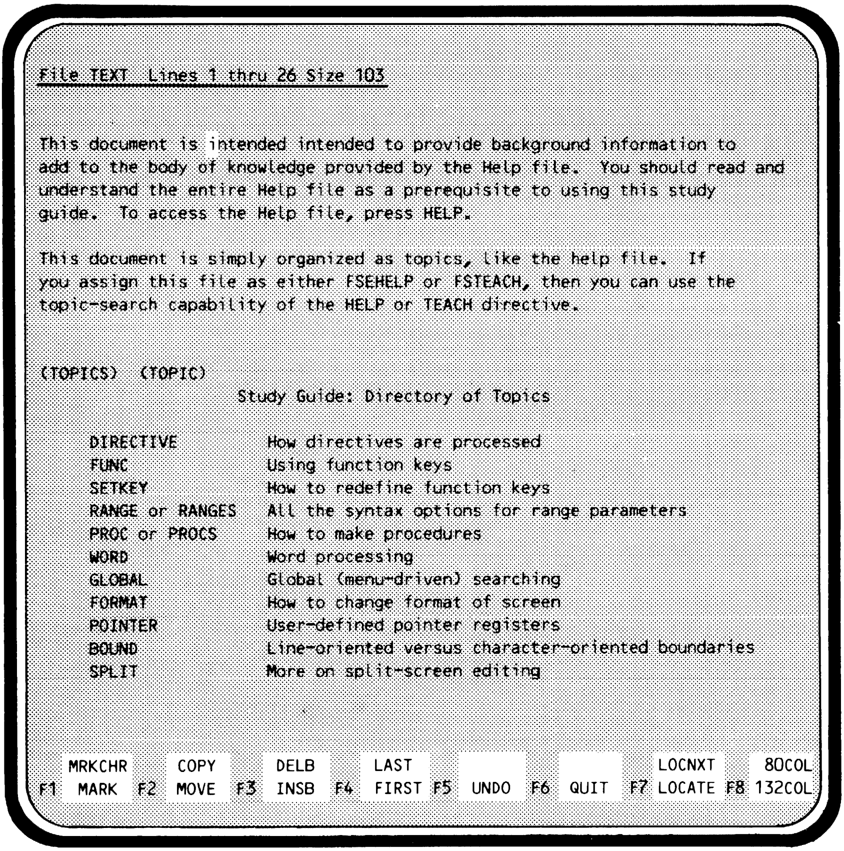
Deletes characters within a line without deleting the line. Its exact function depends on where the cursor is positioned.

If the cursor is on a blank, DELETE WORD deletes that blank and any blanks immediately after it.

If the cursor is on an alphanumeric character, DELETE WORD deletes that character, any alphanumeric characters immediately after it, and any blanks immediately after that.

If the cursor is on any other kind of character, only that character is deleted.

For example, you want to delete the first occurrence of the word *intended* on the following screen. Position the cursor at the *i* in *intended.*



```
File TEXT  Lines 1 thru 26 Size 103


This document is intended intended to provide background information to
add to the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.  To access the Help file, press HELP.

This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH directive.


(TOPICS)  (TOPIC)
                    Study Guide: Directory of Topics

     DIRECTIVE          How directives are processed
     FUNC               Using function keys
     SETKEY             How to redefine function keys
     RANGE or RANGES    All the syntax options for range parameters
     PROC or PROCS      How to make procedures
     WORD               Word processing
     GLOBAL             Global (menu-driven) searching
     FORMAT             How to change format of screen
     POINTER            User-defined pointer registers
     BOUND              Line-oriented versus character-oriented boundaries
     SPLIT              More on split-screen editing


   MRKCHR     COPY      DELB      LAST                          LOCNXT      80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7  LOCATE F8  132COL
```

Then press:

HOME

and enter:

```
D W
```

The first occurrence of the word intended is deleted.

```
This document is░intended to provide background information to
add to the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.  To access the Help file, press HELP.
```

For information on deleting one character at a time, refer to . *(Dot)* later in this section.

IN tab

Deletes a specified tab field. For example, if you set tabs to 1, 7, and 72,

```
D IN 1
```

deletes columns 1 through 6 in the current line and shifts the remaining text left 6 columns.

QUIET

Instructs FSE not to display the results of a deletion.

FSE

Specifies a different file to edit. The cursor is positioned at the first line of the file. If you were previously editing the file, the line you were working on is moved to the top of screen.

Without Parameters

You must specify a file parameter on the FSE directive.

Format

FSE *file charset GET READ SPLIT*

Parameters

file

Specifies the name of the file you want to edit. Any NOS file name is valid.

If you do not specify the GET or READ parameters, FSE uses its latest copy of the file. If the file does not exist, FSE creates a new file with the specified name.

charset

Specifies the character set to be used by FSE. Enter one of the following:

Entry	Character Set Code
`DISPLAY`	6-bit display code (default).
`NORMAL`	6-bit display code.
`ASCII`	6/12-bit display code (ASCII 95-character set).
`ASCII8`	7-bit ASCII code, right-justified in a 12-bit byte.

GET

Instructs FSE to retrieve a permanent file and refreshes the editor workspace. A GET is performed for indirect access files; ATTACH is performed for direct access files. Any changes you have made that aren't permanent will be lost. If there is no permanent copy of the file, FSE reads a local copy.

READ

Refreshes the file image in the editor workspace without a GET or ATTACH operation. Whatever is local (indirect access) or attached (direct access) is read. All changes that aren't permanent are lost.

SPLIT

Divides the screen horizontally to allow editing of two different files. The current file is displayed in the upper half of the screen. The file specified with the FSE directive is displayed in the lower half. You can split one file in order to edit different parts of it. To do this, specify the current file name in the FSE directive. If the two displays overlap the same text, changes will be made to both simultaneously.

The SET VIEW SPLIT directive allows you to specify the number of lines in each portion of the screen; the default is half the screen.

GET

Either displays information on the files you are editing or lists the column numbers.

Without Parameters

You must specify either STATUS or ALIGN.

Format

GET *STATUS ALIGN*

Parameters

STATUS

Displays information on the files you are editing. Following is an example of the information you receive when you enter `GET STATUS` in screen mode.

```
                                     <--PRESS NEXT TO CONTINUE
EDITOR STATUS INFORMATION:

FILES:    NAME    LOCK CHANGE  CHARSET NUMBERED  SIZE POSITION
          FILE1    NO     YES   DISPLAY  YES       293       1
          FSEPROC  NO     NO    DISPLAY  NO         47       1
          FSEHELP  YES    NO    ASCII    NO        542      34
(CURRENT) TEXT     NO     YES   ASCII8   NO        103       9

TAB COLUMNS: 7  72
TAB CHARACTER: \   WARNING WIDTH: 160    PARAGRAPH MARGINS: 1 65 5

FUNCTIONS: KEY   LABEL   DIRECTIVES              KEY    LABEL    DIRECTIVES
            F1    MARK   SM                 SHIFT F1    MRKCHR   SMW
            F2    MOVE   MMTP               SHIFT F2     COPY    CMTP
            F3    INSB   IBP                SHIFT F3     DELB    DB
            F4    FIRST  PF                 SHIFT F4     LAST    VL
            F5    UNDO   UNDO
            F6    QUIT   QUIT
            F7   LOCATE  L/&?/              SHIFT F7    LOCNXT   LN
            F8   132COL  SVC132             SHIFT F8     80COL   SVC80
            F9   MIDDLE  V
            F10  ENDLIN  .END
            F11   SPLIT  .S
            F12   JOIN   .J
            F13   PARA   .F
            F14   COPY   CMTP
            F15  CENTER  .C
            F16
```

Field	Description
NAME	Local files you have been editing. The current file is prefixed with (CURRENT).
LOCK	Read-only status. YES means a read-only file: you cannot make permanent changes. NO means that you can make permanent changes.
CHANGE	Specifies whether you made any changes to the file. YES means you have; NO means you have not.
CHARSET	Character set currently in effect for the specified file. Possible entries are: DISPLAY ASCII ASCII8
NUMBERED	Line number status. YES means it is a sequenced file; NO means it is not a sequenced file.
SIZE	Total number of lines in the file.
POSITION	Line number of the current cursor position in the file. For local files not currently being edited, this is the position at which the cursor will be located when you reenter the file.
TAB COLUMNS	Column numbers of the current tabs.
TAB CHARACTER	Current tab character. Default is \.
WARNING WIDTH	Current line length limit.
PARAGRAPH MARGINS	Current margins set for use by the .FILL and .CENTER directives [refer to the . *(Dot)* directive]. The first two numbers are the left and right margins; the third number is the column in which the first line of a paragraph will begin.
FUNCTIONS	Current labels and directives associated with the programmable function keys.

When screen editing, press:

NEXT

to return to the previous screen.

ALIGN

Writes the column numbers of the current line over the text on the line. The alignment information has no effect on the line over which it is written. The following example shows the information you receive when you enter `GET ALIGN`.

```
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
12345678901234567890123456789012345678901234567890123456789012345678901234567890
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
```

When you are screen editing, the alignment information is written on the current line and stays on the screen until that part of the screen is rewritten.

HELP

Displays the FSE HELP file, which contains descriptions of the FSE directives and their parameters.

Without Parameters

If a directive is not specified, the cursor is positioned at the first line of the HELP file.

Format

HELP *directive*

Parameter

directive

Specifies the name of a directive; the cursor is positioned at the first occurrence of the directive in the HELP file.

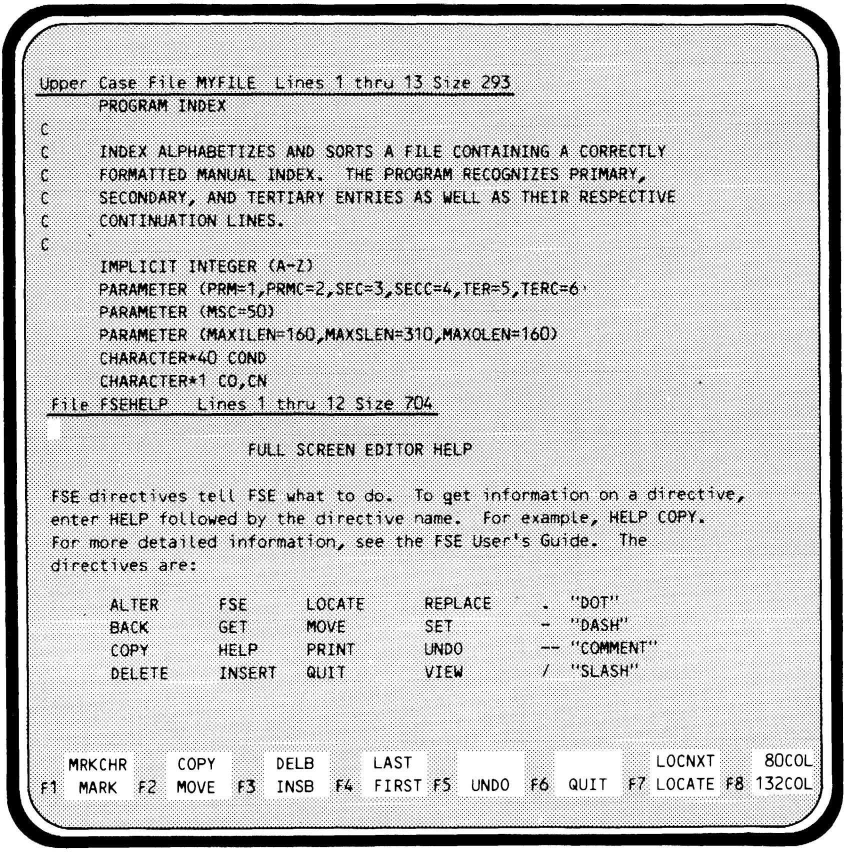
For example, to locate information on the REPLACE directive, enter:

```
H REPLACE
```

The cursor is then positioned at the description of the REPLACE directive within the HELP file. You can also abbreviate the directive you want help with. For example, HELP REPLACE can be entered as:

```
HR
```

When you enter the HELP directive (or press HELP) while screen editing, the screen is split into two equal parts, with the HELP file displayed on the lower half of the screen.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
 File FSEHELP  Lines 1 thru 12 Size 704

                    FULL SCREEN EDITOR HELP

 FSE directives tell FSE what to do.  To get information on a directive,
 enter HELP followed by the directive name.  For example, HELP COPY.
 For more detailed information, see the FSE User's Guide.  The
 directives are:

      ALTER     FSE      LOCATE     REPLACE     .   "DOT"
      BACK      GET      MOVE       SET         -   "DASH"
      COPY      HELP     PRINT      UNDO        --  "COMMENT"
      DELETE    INSERT   QUIT       VIEW        /   "SLASH"


   MRKCHR      COPY      DELB      LAST                              LOCNXT      80COL
F1 MARK    F2 MOVE   F3 INSB   F4 FIRST  F5 UNDO   F6 QUIT   F7 LOCATE  F8 132COL
```

Remarks

When you use the HELP directive while line editing, FSE not only accesses the HELP file, but also prints several lines of text. You are then positioned in the HELP file so you can display more text with PRINT directives.

By default the HELP file is a direct access, public, read-only, permanent file. The default name is FSEHELP from user name LIBRARY. You can, however, assign any file you want to be the default HELP file. To do this, create a local file named FSEHELP and, when the HELP directive is entered (or the HELP key is pressed) your local copy of FSEHELP is used as the default HELP file.

To exit the HELP file, either use the BACK directive or press BACK .

INSERT

Inserts text.

Without Parameters

When you are screen editing and enter the INSERT directive without parameters, you are prompted with the following message.

```
<--- INSERT WHAT?
```

You then enter the text you want to insert. This text will appear as a line after the current line.

When you are line editing and enter the INSERT directive without parameters, you are prompted to enter new lines of text until you enter either a line with a tab character at the end or an empty line (NEXT only.) In the following example, the current line is line 31.

```
?? I
32?inserted line 1
33?inserted line 2
34?inserted line 3
35? NEXT                    (The user ends the insert.)
??
```

If you want the inserted line to begin at a certain tab, include the appropriate number of tab characters before the text you insert. For example, if tabs are set at columns 7 and 72 and you want the inserted line 32 to begin at column 7, enter:

```
32?\inserted line 1
```

\ is the tab character. Refer to the SET CHAR directive for information on the soft tab character.

Format

INSERT *line PREVIOUS string BLANK WORD*

Parameters

line

The lines affected by the insert. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

PREVIOUS

Specifies that you want the insertion before a character or line rather than after.

string

The text you want to insert. When you specify a string, the string is inserted as a single line after the current line. This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.

BLANK

When you are screen editing, inserts blank lines after the current line (useful for entering large amounts of text). Lines of text are left at the top and bottom of the screen. When you are line editing, INSERT BLANK inserts nine blank lines in front of the current line. To remove excess blank lines, position the cursor at the first blank line you want to eliminate and use DELETE BLANK (all subsequent blank lines are deleted also).

WORD

Inserts 30 blank characters at the current cursor position. When you are screen editing, this is sometimes more convenient than repeatedly pressing INSRT . To delete excess blank characters position the cursor at the first blank character you want to remove and enter:

```
DELETE WORD
```

All subsequent blank characters are deleted also.

LOCATE

Locates a specified character string.

Without Parameters

If you enter LOCATE without parameters, the last LOCATE string you specified is used. If you have not previously entered a LOCATE directive, you are prompted to enter the text you want to locate.

Format

LOCATE *WORD direction string range IN tab UPPER QUIET*

Parameters

WORD

Instructs FSE to search for a word (defined as the specified string when it is contained within either nonalphanumeric characters or blanks).

If you are line editing, it positions the cursor on the line containing the string.

If you are screen editing, it positions the cursor at the first character in the found string. For example, suppose you want to locate the word *topic* in the following text.

```
  ░This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.  You should ensure
that your topic comes from the list of topics listed in this file.
```

Press:

HOME

and enter:

```
L W/topic
```

FSE positions the cursor at the first character of the specified string.

```
    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.  You should ensure
that your topic comes from the list of topics listed in this file.
```

Note that FSE passed the word topics in the first line because it is plural. It would not have done so had you not specified WORD.

direction

Specifies the direction LOCATE is to move in the file to find the specified string. FSE searches forward by default, beginning at the current cursor position.

(This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

When you specify the NEXT parameter, FSE begins its search at the next occurrence of the string.

When you specify PREVIOUS, FSE searches backward, beginning at the current cursor position.

The number used with the NEXT, PREVIOUS, and REPEAT parameters refers to the number of lines located, not the number of strings. For example,

```
L N 2 /abc/
```

means locate the next two lines containing abc, no matter how many times abc occurs on each line.

If you do not specify a number,

```
LOCATE NEXT/abc/
```

finds the next occurrence of abc, and

```
LOCATE PREVIOUS/abc/
```

finds the previous occurrence of abc (even if it appears on the current line). For example, you are positioned at line 36 when line editing.

```
36 This is the first and the last of the located lines
```

You then enter:

```
L N /the/
```

You remain at line 36 because the line contains more than one occurrence of the string *the*.

string

Specifies the string you want to locate. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

The following directive locates the string abc.

```
L/abc
```

If the search is successful, LOCATE moves the cursor to the first character of the located string. If the search is unsuccessful, the cursor is not moved and the message `NOT FOUND` is displayed.

If the line located is on the screen already, the cursor is simply positioned at that line. Otherwise, the screen is rewritten with the located line at the top. If you prefer to display the located line in the middle of the screen, follow the LOCATE directive with a VIEW directive.

```
L/abc/;V
```

If you do not specify a string, FSE locates the string on the most recently specified LOCATE directive. If you have not previously entered a LOCATE directive in your editing session, you are prompted with:

```
<--- LOCATE WHAT?
```

You then enter the string you want to find.

To save effort when searching for a long string contained on one line, you can use the ellipsis search technique. If, for example, you want to locate the following string:

```
I will inform you next Monday if a meeting is necessary.
```

you can enter:

```
L/I/../ary./
```

rather than specifying the entire string. You are then positioned at the first character of the specified string.

range

Specifies the lines in which you want FSE to search for the specified string. This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.

When you are screen editing and locate more than one occurrence of a string, FSE displays a directory of all the lines affected. For example, if you were to enter:

```
L A/PARAMETER/
```

the screen might display:

```
                              <-- ENTER LINE NUMBER OR PRESS NEXT
File MYFILE   Displaying located lines
     9        PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
    10        PARAMETER (MSC=50)
    11        PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
    62           COND= 'INVALID PARAMETER - ' // PNAME
    69        STOP 'INVALID PARAMETER.'
```

You could then press:

NEXT

either to view another page of affected lines or, if there are no more pages, to position the cursor in the file at the last line affected.

If you either enter a line number on the directive line or position the cursor at a line in the directory and then press:

NEXT

the cursor is positioned at that line in the file.

IN tab

Locates a specified string appearing within a field bounded by tabs (tab represents the tab field). For example, if tabs were set at columns 5, 20, and 40, the tab fields would be:

Tab Field	Columns
1	1 through 4
2	5 through 19
3	20 through 39
4	40 through the end of the line

To locate all occurrences of the string ABC in columns 20 through 39 (tab field 3), you would enter:

```
L A/ABC/IN 3
```

Refer to the SET directive description for information on setting tabs.

UPPER

Instructs FSE to search the file as if it were all capitalized. For example, if you entered:

```
L U/the/N 3
```

FSE might display:

```
14  The first occurrence
43  THE second occurrence
87  the third occurrence
```

The UPPER parameter is the default setting for display code files. For ASCII files, you must enter the exact string you want to locate.

QUIET

Instructs FSE not to display the located lines and positions the cursor at the last line located.

Remarks

If LOCATE is unsuccessful from within an FSE procedure, no more directives are read from that procedure line; the procedure continues at the start of the next line in the procedure. You can use this to conditionally execute parts of a procedure.

Examples

The following entries are all valid LOCATE directives.

Entry	Meaning
`L F/abc`	Locates first occurrence of abc in the file.
`L\ab/c`	Locates the string ab/c.
`L 20 50/abc`	Locates all occurrences of abc in lines 20 through 50.
`L/abc/20 50`	Locates all occurrences of abc in lines 20 through 50.
`L A/abc`	Locates all occurrences of abc in the edit file.
`L/abc/../xyz/`	Locates the text string that begins with abc and ends with xyz on the same line.
`L A/abc/IN 2`	Locates all occurrences of abc between the first and second tab (tab field 2).
`L P 3/abc`	Locates the previous three lines containing abc; positions the cursor at the first occurrence of abc on the third line.
`L N`	Locates the next line in which the previously specified string occurs.
`L W/boat`	Word-oriented search; finds the next occurrence of the word boat.

MOVE

Moves text from one place to another within a file or between two files.

Without Parameters

If you do not specify a parameter, nothing happens.

Format

MOVE *range (file1) TO line (file2) QUIET*

Parameters

range

Specifies the lines to be moved. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

(file1)

Specifies the file containing the lines to be moved. It can be any NOS file name. The default is the current file.

TO

Separates the lines to be moved from their destination.

line

Specifies the line after which the moved lines will appear. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

This parameter can designate a line either from the current file or from another file. The default is the last line of the file.

For example, to move lines 45 through 50 immediately after line 20, enter:

```
M 45 50 TO 20
```

When you are line editing, FSE displays the moved lines with their new line numbers.

```
21         READ(PVAL,'(I1)') TAB(3)
22      ELSE IF(PNAME.EQ.'SEP') THEN
23         READ(PVAL,'(I1)') SEPCNT
24         IF(SEPCNT.LT.2 .OR. SEPCNT.GT.5) THEN
25            COND='SEP MUST BE 2, 3, 4, OR 5.'
26            PARMERR=.TRUE.
```

When you are screen editing, the cursor is positioned at the first moved line. If you use the F2 MOVE key when screen editing, the lines are moved before, rather than after, the indicated line, because F2 MOVE executes the following directive.

```
M M TO P (MOVE MARK TO PREVIOUS)
```

(file2)

Specifies the file to which the lines are to be moved; used only if the destination is other than (file1). Any NOS file name is valid.

For example, the following directive moves lines 20 through 50 of file MYFILE to the end of file Y.

```
M 20 50 (MYFILE) TO L(Y)
```

The default is the current file.

QUIET

Instructs FSE not to display the changes, only the resulting text.

PRINT

Either prints (displays) a range of lines (line editing) or positions the cursor (screen editing).

Without Parameters

If you do not specify a parameter, the current line is printed.

Format

PRINT *range QUIET*

Parameters

range

Specifies either the lines to be printed (line editing) or the position of the cursor (screen editing). (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

For example, to print the next three lines when line editing, you would enter:

```
P N 3
```

If you were positioned on line 30, lines 31 through 33 would be printed.

```
31    5    ERR=0
32         PARMERR=.FALSE.
33         CALL GETPARM(PNAME,PVAL,ERR)
```

To print the current line (line 30) plus the next two, you would enter:

```
P R 3
```

Lines 30 through 32 would be printed.

```
30         LOGICAL PARMERR
31    5    ERR=0
32         PARMERR=.FALSE.
```

When you are screen editing, rather than printing the lines, the PRINT directive positions the cursor at the last line of the range you specified.

The following example illustrates the difference between specifying the PRINT directive when line editing and when screen editing.

When line editing, if you specify:

```
P 10 15
```

lines 10 through 15 are printed.

```
10        PARAMETER (MSC=50)
11        PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
12        CHARACTER*40 COND
13        CHARACTER*1 CO,CN
14        CHARACTER*10 SLANTS
15        CHARACTER*12 FMTST
```

When screen editing, if you specify the same directive:

```
P 10 15
```

the cursor is positioned at line 15.

```
CHARACTER*40 COND
CHARACTER*1 CO,CN
CHARACTER*10 SLANTS
CHARACTER*12 FMTST
CHARACTER*(MAXILEN) INPLIN
CHARACTER*7 PVAL,PNAME
CHARACTER*7 INPFILE,OUTFILE
CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
LOGICAL PARMERR
```

When screen editing, think of the PRINT directive as a positioning directive, because it positions the cursor to a specified line. The following PRINT directives may prove useful during screen editing.

`P L` Positions you to the last line of the file.

`P F` Positions you to the first line of the file.

`P 1` Positions you to the first line of the file.

QUIET

Instructs FSE not to print the specified lines, but instead to position the cursor at the last line specified.

QUIT

Exits FSE or an FSE procedure.

Without Parameters

If you enter QUIT without parameters, you exit FSE, changing only the local copies of your files. The exception is when you have attached a direct access file in write mode. In this case, entering QUIT without parameters makes your changes permanent.

Format

QUIT *REPLACE QUIET UNDO PROC /NOS command*

Parameters

REPLACE

Instructs FSE to make your changes permanent by defining, replacing, or copying to permanent storage as necessary. If you have created a new file within FSE, use the REPLACE parameter to make it permanent.

The REPLACE and UNDO parameters cannot be included in the same QUIT directive.

QUIET

Instructs FSE not to print the file status messages when it is stopped.

UNDO

Undoes all changes made during your current editing session. If you specify the UNDO parameter, all changes made since the last time you entered FSE are lost. It also becomes impossible to return to the point at which you stopped FSE during this editing session.

The REPLACE and UNDO parameters cannot be included in the same QUIT directive.

NOTE

The UNDO parameter is the only parameter in FSE that must be completely spelled out.

PROC

Used only in FSE procedures; instructs FSE to stop processing the current procedure without exiting FSE. QP (QUIT PROC) can be used without parameters in place of an EOR to separate procedures. Ordinarily QP should be by itself at the start of the line. Exceptions may occur in conditional procedures. (For examples of conditional procedures, refer to *Conditional Procedures* in section 6.)

/NOS command

Performs the function specified by the NOS command you enter. For example, to stop FSE, replace changed files, and reenter FSE, enter:

```
Q R/FSE
```

Anything on the line following the / is considered part of the NOS command.

Remarks

When you exit FSE using the QUIT directive, a message of the following type appears.

```
FILE: filename (status)
```

Field	Description	
filename	Name of a file you have accessed during your editing session.	
status	May be any of the following.	
	LOCAL	Either you have created a file and have not made it permanent (it will disappear when you log off NOS) or you have made changes to a permanent file and have not made the changes permanent (the changes you made will not be there the next time you access the file). To make a file (or changes to a file) permanent, enter the QUIT REPLACE directive (QR).

Field	Description	
	NO CHANGES	No changes have been made to the file.
	PERMANENT	You have replaced the permanent copy of the file with the changed local copy.
	READ-ONLY FILE	You may not make any changes to the permanent copy of the specified file.
	If no message appears following a file name, you have made changes only to the local copy of a file; the changes are not permanent.	

For example, if you entered the QUIT REPLACE directive to stop FSE while editing file MYFILE, the following messages might appear.

```
FILE:  MYFILE  (PERMANENT)
FILE:  FSEPROC (NO CHANGES)
FILE:  FSEHELP (NO CHANGES) (READ-ONLY FILE)
```

The PERMANENT message tells you that the changes you made to file MYFILE are now permanent. This message appears when you use the REPLACE parameter.

File FSEPROC is always one of the files FSE accesses. In this example, the NO CHANGES message means you have not made any changes to file FSEPROC.

File FSEHELP is the online help file that is made available to FSE when you enter the HELP directive or press HELP. The READ-ONLY FILE message means you cannot make any permanent changes to the file.

If you do not want these messages displayed when you stop FSE, include the QUIET parameter.

REPLACE

Replaces one text string with another.

The REPLACE directive works very much like the LOCATE directive. The difference is that the REPLACE directive locates the string you specify and then replaces it with another string (that you also specify).

Without Parameters

If you do not specify a parameter, REPLACE uses the last string specified and executes the directive. If you have not previously specified a REPLACE string, you are prompted for the intended text.

Format

REPLACE *WORD direction string1 string2 range*
 IN tab UPPER QUIET

Parameters

WORD

Instructs FSE to search for and replace a word (word is defined as string1 when it is contained within either nonalphanumeric characters or blanks). For example, to replace the word *topic* with the word *subject* in the following text:

```
    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.  You should ensure
that your topic comes from the list of topics listed in this file.
```

press:

(HOME)

and enter:

```
R W/topic/subject/
```

FSE replaces the first occurrence of the word topic with the word subject.

```
    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
subject-search capability of the HELP or TEACH command.  You should ensure
that your topic comes from the list of topics listed in this file.
```

Notice that FSE passed the word topics in the first line, because it is plural. It would not have done so, had you not specified WORD.

direction

The direction REPLACE is to move in the file. (This is a common parameter. For a listing of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

FSE searches forward by default, beginning at the current cursor position.

When you specify NEXT, FSE begins its replacement at the next occurrence of string1, in effect skipping an occurrence.

When you specify PREVIOUS, FSE searches backward, beginning at the current cursor position and skipping the last string1 located.

Using a number greater than 1 with the NEXT, PREVIOUS, and REPEAT parameters specifies the number of lines to search (rather than the number of occurrences to replace).

For example:

```
R N 2 /abc/xyz/
```

searches the two most current lines and replaces all occurrences of abc with xyz.

```
R N /abc/xyz/ or R N 1 /abc/xyz/
```

searches for and replaces the next occurrence of abc with xyz.

```
R C C /abc/xyz
```

replaces all occurrences from one end of the current line to the other.

Examples:

Suppose that the cursor is positioned at the beginning of the following lines.

```
░A line of abc,abc,abc.
 A second line of abc,abc,abc.
```

Press:

HOME

and enter:

```
R N /abc/xyz/ (or R N 1 /abc/xyz/)
```

and you get:

```
A line of xyz,abc,abc.
A second line of abc,abc,abc.
```

Enter:

```
R N 2 /abc/xyz/
```

and you produce:

```
A line of xyz,xyz,xyz.
A second line of xyz,xyz,xyz.
```

Enter:

```
R C C /abc/xyz/
```

and the result is:

```
A line of xyz,xyz,xyz.
A second line of abc,abc,abc
```

string1

The string you want replaced. (This is a common parameter. For a list of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

If you do not specify string1, FSE uses string1 on the most recently specified REPLACE directive. If, however, you have not previously entered a REPLACE directive in your editing session, you are prompted to enter the intended text.

To replace a long string contained on a line, use the ellipsis search technique, which allows you to replace a long string without typing it out entirely. For example, you want to replace the string:

```
I will inform you next Monday if a meeting is necessary.
```

with the string:

```
No meeting is necessary.
```

Rather than specifying the entire first string, enter:

```
R/I/../ary./No meeting is necessary./
```

The first string is then replaced with the second.

If FSE cannot find string1, it displays the message `NOT FOUND`. If FSE cannot find a string1 from within an FSE procedure, no more directives are read from that procedure line (if the line contains more than one directive); execution continues at the start of the next line in the procedure.

string2

The string you want to substitute for string1. (This is a common parameter. For a list of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

For example, suppose the cursor is positioned at the line:

```
I iill inform you next Monday if a meeting is necessary.
```

and you want to replace the first i with w, enter:

```
R/i/w
```

FSE locates the first line containing i and replaces the first i with w. The following line results:

```
I will inform you next Monday if a meeting is necessary.
```

When you are screen editing, if the line affected by REPLACE is on the screen, the cursor is positioned at that line. Otherwise, the screen is rewritten with the affected line at the top. If you prefer to display the affected line in the middle of the screen, follow the REPLACE directive with a VIEW directive, as follows:

```
R/abc/xyz/ ; V
```

range

Specifies the lines to be affected by the REPLACE directive. (This is a common parameter. For a listing of all possible entries, refer to the *Common Parameter Index* on the inside front cover.)

When you are screen editing and replace more than one occurrence of a string, FSE displays a directory of all the lines affected. For example, if you were to enter:

```
R A/PARAMETER/PARAMETER/
```

the screen might display:

```
                                      <-- ENTER UNDO OR PRESS NEXT TO END
File MYFILE   Displaying changed Lines
      9       PARAM (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
     10       PARAM (MSC=50)
     11       PARAM (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
     62          COND= 'INVALID PARAM - ' // PNAME
     69       STOP 'INVALID PARAM.'
```

Press:

NEXT

either to view another page of affected lines or, if there are no more pages, to position the cursor in the file at the last line affected.

If you enter a line number on the directive line and press:

NEXT

the cursor is positioned at that line in the file. If you either enter:

```
UNDO
```

or press:

F5 UNDO

the REPLACE operation is undone.

IN tab

Replaces text appearing only in certain tab fields. tab represents a tab field; for example, if tabs are set at columns 5, 20, and 40, the tab fields are:

Tab Field	Columns
1	1 through 4
2	5 through 19
3	20 through 39
4	40 through the end of the line

To replace all occurrences of abc with xyz in columns 20 through 39 (tab field 3), enter:

```
R A/abc/xyz/ IN 3
```

Refer to the SET directive description for information on setting tabs.

UPPER

Instructs FSE to search the file as if it were all uppercase, that is, capitalization is ignored. If, for example, you entered:

```
R U/abc/xyz/N3
```

REPLACE would replace each of the following lines with xyz.

```
14  ABC
43  Abc
87  abc
```

The UPPER parameter is always in effect for display code files.

QUIET

Instructs FSE not to display the lines changed, only the results of the changes; positions the cursor at the last line replaced.

Examples

The following are all valid REPLACE directives.

Entry	Meaning
`R/abc/xyz`	Replaces next occurrence of abc with xyz.
`R A/abc/xyz`	Replaces all occurrences of abc in the file with xyz.
`R\ab/c\c/pm`	Replaces the string ab/c with the string c/pm.
`R 20 50/abc/xyz`	Replaces all occurrences of abc in lines 20 through 50 with xyz.
`R/abc/../xyz/1234/`	Replaces text string beginning with abc and ending with xyz with the string 1234.
`R/abc/xyz/IN 2`	Replaces all occurrences of abc with xyz in tab field 2.
`R N 3/abc/xyz`	Replaces all occurrences of abc with xyz on the next 3 lines (including the current line).
`R/abc//`	Deletes the next occurrence of abc.
`R//xyz/`	Replaces the previously specified REPLACE string with xyz.
`R`	Uses both previously specified strings and performs another REPLACE.
`R/abc/xyz/M`	Replaces abc with xyz on marked lines.
`R/abc/xyz/S`	Replaces abc with xyz on all lines currently on the screen.

SET

Sets various FSE and file parameters.

Without Parameters

You must enter a keyword after the SET directive.

Format

SET keyword

Parameters

There are many valid keywords that you can use with the SET directive. The following list of keywords is in alphabetical order.

ANNOUNCE/string/

Enables you to display prompts or messages, usually from within FSE procedures. The string you specify is displayed in the FSE message area before the next input is accepted.

CHAR character

Sets the specified punctuation character as the soft tab character.

If you want to specify the semicolon, use SEMI.

If you want to specify the blank character, use BLANK.

If you want no soft tab character (as is often appropriate when screen editing), specify SET CHAR either with no parameter or followed by a semicolon.

FILENAME file

Changes the name of the current file. Using this parameter, you can change a file and then have FSE copy the changed version to a new file, keeping the original file intact.

HEADER value

Tells FSE how to display the title line. If value is YES, FSE displays the entire title line (also the default). If value is NO, FSE abbreviates the title line to the file name.

INCREMENT number

Sets the line number increment for lines inserted into BASIC or interactive FORTRAN sequenced files. The default increment is 100. For information on sequenced files, refer to the NUMBER keyword.

KEY number SHIFT string LABEL string

Redefines a programmable function key to execute the directive or directives you specify with the string parameter.

number is a number from 1 through 16 (the numbers of the function keys).

Including the SHIFT parameter defines the shifted function key (you can define up through 32 functions).

To separate directives, use a semicolon. The following SET KEY directive redefines function key F1 to execute the LOCATE NEXT and VIEW directives.

```
S K  1/L N;V
```

To return a function key to its default setting, enter only the number parameter.

```
S K  1
```

If you do not include the LABEL and string parameters, the key label is the same as the directive string. If you do not include the directive string, but do include the LABEL and string parameters, the label changes but the directive executed by the key remains the same. The directive string can be 244 or fewer characters; the label string can be 6 or fewer characters. For example, the following default function keys for the Viking 721 terminal are set as follows:

```
S K  1/S M/L/MARK/

S K  2/M M T P/L/MOVE

S K  S 2/C M T P/L/COPY/
```

Function key F2 MOVE can be redefined to move lines after instead of before the current line by entering:

```
S K  2/M M/
```

You can include blanks in a function key label to make the label more readable. A good rule is to leave a space before labels of five or fewer characters.

LINE

Sets line mode for the current editing session; it does not affect other editing sessions or other parts of your terminal session. Line mode is set either by default or by entering a NOS LINE command before starting FSE.

MARK range WORD

Sets one or two temporary markers to be used with another directive.

When you are screen editing, the SET MARK directive performs the same function as the F1 MARK key, while the SET MARK WORD directive performs the same function as F1 MRKCHR. Refer to section 2, *Screen Editing*, for an example using the F1 MARK key.

By default, SET MARK marks an entire line or range of lines. You can then refer to these lines in a later directive by using the MARK parameter. For example,

```
S M  5 10
```

marks lines 5 through 10. Later, the directive:

```
C M  T 50
```

copies lines 5 through 10 immediately after line 50.

SET MARK is frequently used in screen editing to mark the current line for later copying, moving, and so forth. A single SET MARK directive marks the current line. A second SET MARK directive marks the current line and all lines between the first and second marks.

The WORD parameter has two effects. The first causes characters to be marked instead of lines. The second causes any later operations using the MARK parameter to perform character-oriented operations instead of line-oriented operations. For example, the word *character* is marked in the following text.

```
The character mark causes later operations to affect only
characters within lines.
```

A DELETE MARK directive results in:

```
The░mark causes later operations to affect only characters
within lines.
```

The UNDO directive first clears any previously set marks.

NUMBER value

Instructs FSE whether to treat a file as sequenced or unsequenced. The values are:

value **Function**

AUTO Instructs FSE to treat the file as if it were unsequenced (default). In line editing, FSE automatically generates line numbers when lines are displayed. These numbers are provided for reference and are not part of the text of the file.

BASIC Instructs FSE to assume that the file is sequenced. If you are creating a file and you number the lines, FSE adds zeros to increase the number to five digits and automatically inserts a blank. For example, when you type `10The first line`, the line is printed as:

```
00010 THE FIRST LINE
```

If you don't number the lines, for example `The first line`, FSE prints only the text (without a line number).

```
THE FIRST LINE
```

If you insert text using the INSERT directive, FSE automatically generates a sequence number. For lines added to the end of a file, it adds 100 to the last number. For example, assume the following lines are the last two lines in a file.

```
00010 THE FIRST LINE
00020 THE SECOND LINE
```

If you position the cursor at line 00020 (the last line) and use the INSERT directive to specify `I the new line`, FSE writes:

```
00010 THE FIRST LINE
00020 THE SECOND LINE
00120 THE NEW LINE
```

For lines inserted between existing lines, FSE adds half the value of the preceding line. If you position the cursor at line 00010 in the preceding example and follow the same procedure, FSE prints:

```
00010 THE FIRST LINE
00015 THE NEW LINE
00020 THE SECOND LINE
```

value	Function
FORTRAN	The same as BASIC except that it doesn't insert a space between the line number and the first character.
NONE	Instructs FSE to treat the file as if it were unsequenced. When line editing, FSE displays lines without generating line numbers.

PROMPT value

Enables you to specify whether you want the programmable function key prompts to be displayed on the screen. Values can be:

value	Description
0	Instructs FSE not to display programmable function key prompts.
1	Instructs FSE to display the first eight programmable function key prompts (default).
2	Instructs FSE to display all 16 programmable function key prompts.

SCREEN

Sets screen editing mode and rewrites the screen if you are already in screen mode.

TAB $col_1\ col_2\ col_3..col_{20}$

Sets the tab stop at the specified column number (col_n). Default tab settings are 7 and 72. You are allowed up through 20 tab settings. If no column numbers are specified, all tabs are cleared. You cannot add tabs to existing tab settings. For example, if you need to add a tab at column 15 between 7 and 72, enter:

```
S T 7 15 72
```

When you are screen editing, FSE sets a hardware tab at column 1, allowing you to use the ⇤ key to position the cursor at the beginning of the line. This does not affect the SET TAB values.

UNDO value

Sets the UNDO feature on or off. Values can be YES or NO. YES enables you to use the UNDO directive as described previously in this section. When you specify NO, you cannot use the UNDO directive. However, with the UNDO feature off, massive changes to files use less system resources and complete more quickly, which is particularly important if you are doing batch processing. The default is YES.

VIEW value

The SET VIEW directive enables you to change the format of your screen. Valid value entries are:

value	Function
COLUMN num	Specifies the number of columns (num) to be displayed. The default is 80 columns. SET VIEW COLUMN is a hardware option, not a software option. It tells your terminal to display the number of columns you specify. For example, on a Viking 721 terminal, SET VIEW COLUMN 81 sets the terminal to 132-column mode, and all 132 columns of text are displayed.
EDIT num	Defines a column width (num) of text that can be edited. The number specified is always the right margin and 1 is always the left margin. For example, to limit yourself to editing only columns 1 through 20, enter 20 as the EDIT value. Any column following column 20 will not be affected by succeeding edit functions.
LINE num	Specifies the number of lines (num) to be displayed. The default is the capacity of your terminal.
OFFSET num	Changes the leftmost column displayed so you can view wide lines. num represents the column number of the new left margin. To return to the default column range of 1 through 72, enter: `S V O 1`

value	Function
SPLIT num	Specifies the number of lines (num) used for the bottom half of a split screen edit. The default is one-half the capacity of your terminal.
WARN num	Specifies a line length limit. The default limit is 160. If, in the course of directive processing, a line is encountered that exceeds the SET WARN limit, directive processing stops and a warning message is displayed.

Refer to *Changing Your Screen's Format* in section 6 for more examples of SET VIEW directives.

WORD value

The SET WORD directive enables you to set character attributes and specify margins. Valid value entries are:

value	Function
CHAR character	Defines the character attribute as alphanumeric or punctuator. The directive reverses the current attribute of the indicated character and issues a status message to the user. A space cannot be redefined. It is treated as a filler and punctuator for the DELETE WORD option. The semicolon can be redefined using the directive: S W C SEMI The character attribute value affects the UPPER and WORD options of the LOCATE and REPLACE directives, the WORD option of the DELETE directive, and the &W micro symbol.

value	Function
	The UPPER option of the LOCATE and REPLACE directives will accept the following pairs of lowercase to uppercase mappings (provided both characters have been defined by a S W C directive as alphanumeric). { [}] ¦ \ ~ ^ \ @
FILL margin1 margin2 margin3	Sets margins other than the default settings (1, 65, and 5) for use with the .CENTER and .FILL directives. margin1 sets the left margin; margin2 sets the right margin; and margin3 sets the paragraph indentation. Refer to section 5, *Sample Screen Editing Session*, for examples of word processing functions.

X line WORD

Y line WORD

Z line WORD

Sets the X, Y, or Z pointer to the specified line value. These pointers can be used in any directive in which a line parameter is allowed. If two pointers are specified in a range parameter, they must be in the same file. These pointers are permanent (as compared with those set by the MARK function) and can be reset at any time.

The WORD parameter sets the pointer to a character position rather than a line value. Any subsequent directives that use a pointer directive are character-oriented rather than line-oriented. Refer to the SET MARK directive for more information.

TEACH

Provides a practice file on which you can try some of the FSE operations. If you have a Viking 721 terminal, the file also includes a tutorial, which guides you through the basic editing functions.

Without Parameters

If you enter TEACH with no parameter specified, the cursor is positioned to the first line of the TEACH file.

Format

TEACH *parameter*

Parameters

The parameters are titles of topics. When you specify a parameter (applicable only to Viking 721 terminals), the cursor is positioned at the first occurrence of the topic in the TEACH file. If you enter an invalid parameter, the cursor is positioned at the beginning of the TEACH file. Valid parameters are:

- BKSPACE
- BKTAB
- BKW
- CENTER
- COPY
- CURSOR
- DLETE
- SHIFTDL
- ENDLIN
- FWD
- HOME
- INSRT
- SHIFTIN
- JOIN
- KEYS
- LOCATE
- LOCNXT
- MARK
- MOVE
- MRKCHR
- NEXT
- PARA
- QUIT
- REPLACE
- SHIFT
- SPLIT
- TAB
- UNDO
- WORDPRO
- 132COL
- F1
- SHIFTF1
- F2
- SHIFTF2
- F5
- F6
- F7
- SHIFTF7
- F8
- SHIFTF8
- F10
- F11
- F12
- F13
- F15

Remarks

To return to the file you were editing, press:

BACK

Unlike the HELP file, the TEACH file is not protected. Its purpose is to teach you how various features work, and you are encouraged to change it as much as you like.

The tutorial accessed for the Viking 721 terminal does not cover every feature of FSE. Its purpose is to acquaint you with the basics of screen editing.

UNDO

Undoes changes made to the current file.

Format

UNDO

Remarks and Examples

Successive UNDO directives work backward in chronological order to restore the file to previous conditions. For example, suppose you made the following changes to the current file in the order shown.

1. Replaced all a's with b's.
2. Moved the first three lines of the file to the end of the file.
3. Changed a line of text by overtyping the correction.

The first time you enter UNDO, the changed line of text is restored to its original form.

The second time you enter UNDO, the three lines now at the end of the file are returned to the beginning of the file.

The third time you enter UNDO, the b's are restored to a's.

If you have pressed F1 MARK or entered the SET MARK directive to mark lines, the first UNDO removes the marks and nothing else. A message is displayed telling you that the marks have been removed. The next UNDO will then work as shown previously.

UNDO can undo your changes if you exit FSE using the QUIT or QUIT REPLACE directives; just enter the FSE command without parameters and enter UNDO as many times as needed.

You can think of the UNDO directive as a type of security blanket. If you can't quite recall how a directive works, give it a try anyway. If you don't like the results, you can undo them.

VIEW

Allows you to view a group of lines.

Without Parameters

If you do not specify a parameter while line editing, the four preceding lines, the current line, and the four succeeding lines are displayed. If you do not specify a parameter while screen editing, the current line is centered vertically on the screen.

Format

VIEW *line direction SCREEN HOME*

Parameters

line

Specifies the line you want to view. (This is a common parameter. For a list of possible entries, refer to the *Common Parameter Index* on the inside front cover.)

direction

Specifies the direction you want to view. The direction parameter can only be NEXT or PREVIOUS (described in *Common Parameters*, earlier in this section).

When you are line editing, the VIEW directive enables you to look at a number of lines at once. The following are valid line mode VIEW directives.

Entry	Meaning
V	Displays the current line, the four preceding lines, and the four succeeding lines.
V N	Displays the current line and the next eight lines.
V P	Displays the previous eight lines and the current line.

Successive VIEW NEXT or VIEW PREVIOUS directives page through the file.

SCREEN

Available only when screen editing, enables you to view the previous or next screen.

HOME

Available only when screen editing, moves the cursor to the FSE directive line without changing your view of the file text.

Examples

When you are screen editing, VIEW is used to position text on the screen. The following are valid screen mode VIEW directives.

Entry	Meaning
V	Vertically centers the current line on the screen.
V N	Positions the current line at the top of the screen; same as the UP key.
V P	Positions the current line at the bottom of the screen. Same as the DOWN key.
V N S	Moves forward one screen; same as the FWD key.
V P S	Moves backward one screen; same as the BKW key.
V H	Moves the cursor to the directive line; same as the HOME key.

. (DOT)

Performs word processing functions according to the specified parameter.

Format

.parameter

Parameters

CENTER

Centers the current line horizontally within preset margins. The default margin settings are 1 and 65. Use the SET WORD FILL directive to change these margins.

DELETE

Deletes the current character.

END

Moves the cursor to the end of the current line.

FILL

Adjusts the words or sentences within a paragraph of text to bring line lengths as close as possible to preset margins. A paragraph of text is a group of lines that does not contain a blank line. Margin defaults are 1, 65, and 5. Use the SET WORD FILL directive to change these margins.

INSERT

Inserts one blank character at the current cursor position.

INSERT/string

Inserts a string at the current cursor position.

JOIN

Combines the current line with the line following it.

POS n

Moves the cursor position to column n of the current line.

SPLIT

Divides one line into two. The line is split at the cursor position.

NOTE

During line editing, only the .CENTER and .FILL directives function visibly as indicated. The other functions work, but you will not see the results until the file is listed. For this reason, they are not very useful for line editing. For more information on word processing functions in line editing, refer to section 7, *Line Editing*.

- (DASH)

Calls an FSE procedure. An FSE procedure is one or more directives in a named record in a file. More information on FSE procedures follows the parameter descriptions.

Format

-procname *(file)*

Parameters

procname

The name of the FSE procedure you want to execute.

(file)

Calls a procedure from a local or permanent file other than the default file FSEPROC. You always have an FSEPROC file; if there is no local file named FSEPROC, a NOS GET command is executed. If there is no indirect access FSEPROC file, a system default FSEPROC file is retrieved from the user name LIBRARY.

To call procedure FIND from local file PROCFIL, enter:

```
-FIND (PROCFIL)
```

FSE goes to file PROCFIL and executes the FIND procedure.

Remarks

An FSE procedure must begin with its name alone on the first line, followed by one or more lines of FSE directives. The procedure must then end with either an end-of-record (EOR) line or a QUIT PROC directive. You cannot leave blank lines or comment lines between procedures. For example, the following two procedures delete all lines containing a specified string.

```
DELALL
S A/DELETE WHAT?/
L F/&?/; -DA
QP
DA
D;L;-DA
(EOR)
```

Assuming these records are in your FSEPROC file, you can call the first procedure by entering:

```
-DELALL
```

The procedure then prompts you with DELETE WHAT?. You enter a string and all lines in the current file containing that string are deleted.

Each directive in a procedure is executed in sequence. If a directive attempts to position the cursor outside the current file or to locate a string that does not exist, FSE stops executing that line of the procedure and continues with the next line. FSE returns to your editing session whenever it encounters (EOR) or a QUIT PROC directive.

Procedure calling is only one-level deep; that is, the - directive jumps to the named procedure, terminating the execution of any current procedure.

The procedure named STARTUP in FSEPROC is executed each time you start FSE, but not when you resume a previous session. You must create your own STARTUP procedure.

When certain two-character sequences, called micros, are contained in FSE procedures or programmable function key strings, FSE interprets them in a particular way. They are not interpreted when they are entered as directives or when they are within your files.

Micro	Interpreted as
&C	The column number of the current character position beginning at 1.
&F	The name of the current file.
&L	The number of the current line beginning at 1.
&T	The current terminal type (721, 722, VT100, Z19, and so forth).
&W	The word on which the cursor is positioned, or, if the cursor is not positioned on a word, the next word to the right.
&n	The nth parameter[†] of a procedure call, for example, &2.
&?	A prompt for you to enter input (your input replaces the &?).
&&	&

For example, suppose you have a procedure in FSEPROC defined as:

```
FCOMP
/FTN5,I=&F,L=0.
QP
```

[†] In a procedure, a parameter consists of either a series of nonblank characters or a string.

To compile the current file as a FORTRAN 5 program, you enter:

```
-FCOMP
```

For added convenience, the procedure can be assigned to a programmable function key. To define the F9 key as -FCOMP, enter:

```
S K 9 /-FCOMP/L/FORT/
```

Whenever you press:

```
F9 FORT
```

the current file is compiled as a FORTRAN 5 program (FCOMP).

Procedure parameters are taken from the directive that calls the procedure. A parameter is a consecutive sequence of nonblank characters or a string. For example, in the following procedure call:

```
-XYZ /hi there/ bye now
```

micro &1 within procedure XYZ is replaced by `/hi there/`; &2 is replaced by `bye`; and &3 is replaced with `now`.

Refer to section 6, *Advanced FSE Functions*, for more examples of FSE procedures.

NOTE

If you use a procedure to redefine a function key with a micro, the ampersand contained in the micro must be written twice. At the time the procedure is executed, the double ampersand is reduced to a single ampersand and stored in the function key. When the function key is pressed, the single ampersand and letter are identified as the correct micro.

Failure to double the ampersand in the procedure results in the micro being processed at start up time rather than when the function key is pressed (that is, you will have redefined the key incorrectly).

-- (COMMENT)

Enables you to include comments following FSE directives.

Format

-- comment

Example and Remarks

comment is the remark you want to make after an FSE directive, for example:

```
L L 2
--  Locate last two procedures
S A/LAST TWO OF WHAT?/      -- Prompt user
L L P 2/&?/                 -- Locate specified string
Q P
```

These comments have no effect.

NOTE

Comments cannot appear on the line containing the procedure name, the QUIT PROC directive, or the (EOR) line.

/ (SLASH)

Exits FSE and executes the characters following the slash as a NOS command.

NOTE

Some NOS commands will not work with the / directive. These include LIB, NOSORT, RUN, the -procname format of the BEGIN command, and primary file editing commands (refer to the *NOS 2 Reference Set, Volume 3*).

Format

/command

Examples and Remarks

command is the NOS command you want to execute. For example, to enter the NOS CATLIST command while you are editing a file, enter:

```
/CATLIST
```

To reenter your editing session at the point you entered the NOS command, enter the FSE command with no parameters.

```
FSE
```

If the / directive is used to execute a NOS procedure that ends with the following command, the procedure automatically reenters when it is finished.

```
REVERT,EX.FSE.
```

The / directive is simply shorthand for the QUIT directive with a string parameter. That is,

```
/command
```

is equivalent to:

```
Q/command
```

You can use the Q/command format to specify other QUIT parameters. For example,

```
QR/FSE
```

replaces all changed files and restarts FSE.

Sample Screen Editing Session 5

Sample Screen Editing Session 5

In this section, sample files will be edited using some of the advanced screen editing functions, including:

- Copying lines between files using split-screen editing.
- Performing word processing functions.
- Changing your screen's format.

Use this section to familiarize yourself with the editing functions available. The editing session does not include all of the advanced functions; for a comprehensive description, refer to section 6, *Advanced FSE Functions*.

NOTE

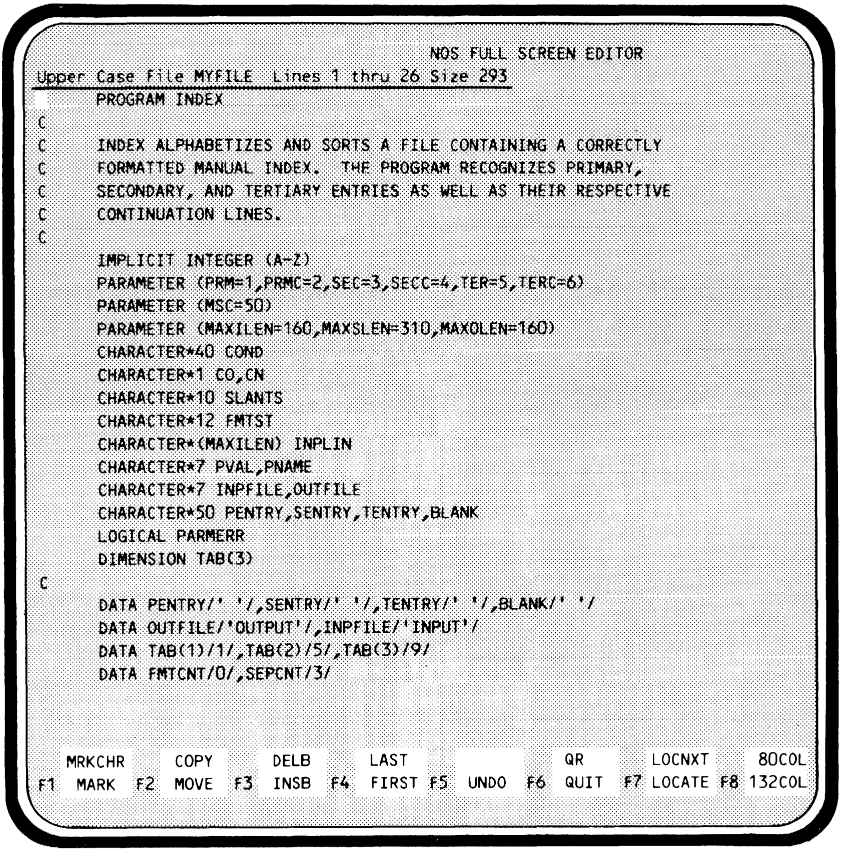
For each task, sample display screens will show you how an imaginary file would look before and after editing. The sample files are not released with the system. You must create your own sample files if you want to perform the tasks in the examples.

Starting FSE

Suppose you want to edit file MYFILE. To start FSE and get your file out of permanent storage, enter:

```
FSE,MYFILE,G
```

The file is displayed on the screen.



```
                                        NOS FULL SCREEN EDITOR
Upper Case File MYFILE  Lines 1 thru 26 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
      CHARACTER*10 SLANTS
      CHARACTER*12 FMTST
      CHARACTER*(MAXILEN) INPLIN
      CHARACTER*7 PVAL,PNAME
      CHARACTER*7 INPFILE,OUTFILE
      CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
      LOGICAL PARMERR
      DIMENSION TAB(3)
C
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/



   MRKCHR     COPY      DELB      LAST                QR      LOCNXT     80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7 LOCATE F8 132COL
```

Notice that the prompt for function key F6 is not the default described in section 2. The default setting:

F6 **QUIT**

has been redefined to include the shifted F6 function QUIT REPLACE, abbreviated QR. (Redefining programmable function keys is described in section 6.)

Copying Lines Between Files

To copy lines from file MYFILE to another file (PARTS), use split-screen editing.

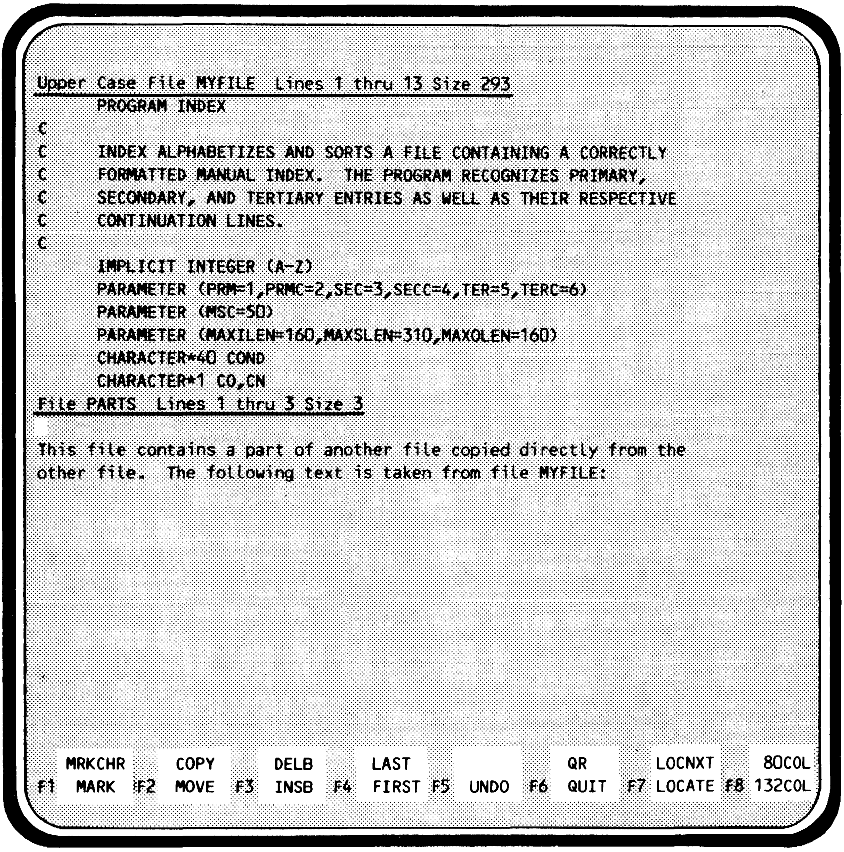
First, display file PARTS on the lower half of the screen by pressing:

(HOME)

and entering the FSE directive:

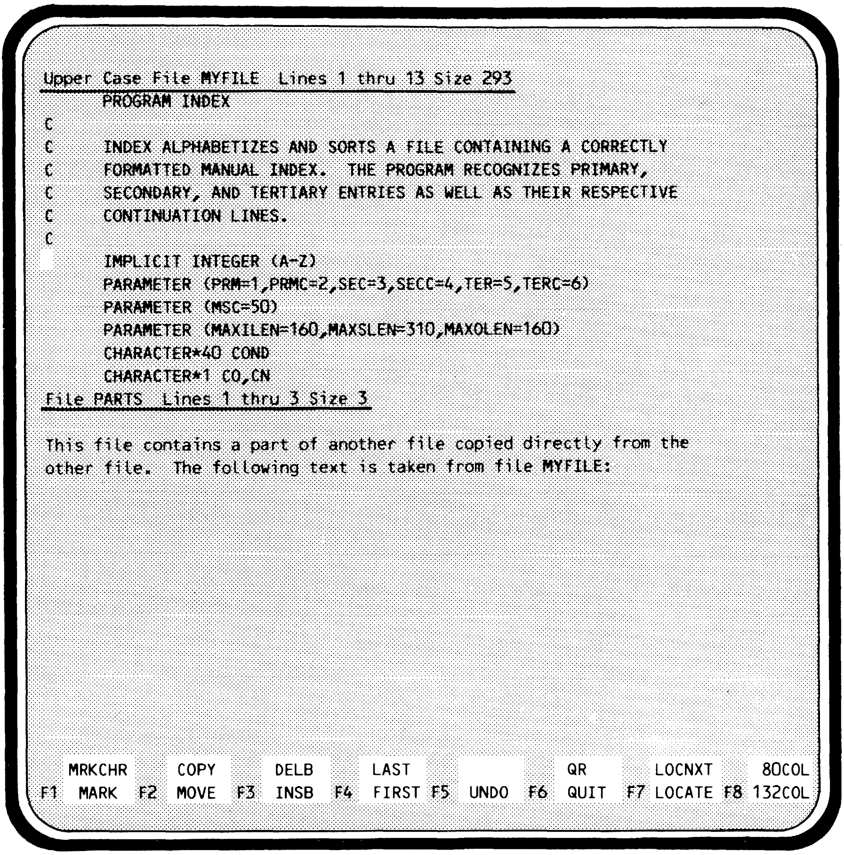
```
F PARTS A S
```

A specifies 6/12-bit ASCII characters, and S specifies a split screen. The following screen results.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File PARTS  Lines 1 thru 3 Size 3

This file contains a part of another file copied directly from the
other file.  The following text is taken from file MYFILE:
```

Then, position the cursor at the first line of the range to be copied.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File PARTS  Lines 1 thru 3 Size 3

This file contains a part of another file copied directly from the
other file.  The following text is taken from file MYFILE:
```

```
  MRKCHR     COPY      DELB      LAST             QR      LOCNXT      80COL
F1 MARK   F2 MOVE  F3 INSB  F4 FIRST F5 UNDO  F6 QUIT  F7 LOCATE F8 132COL
```

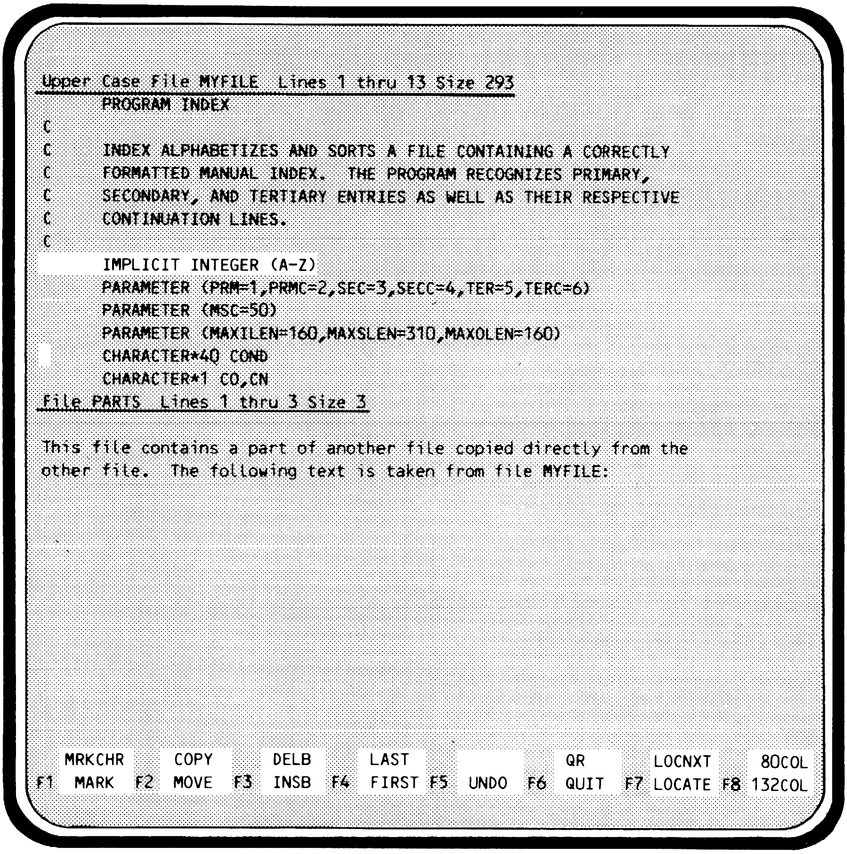
Press:

F1 **MARK**

FSE highlights the marked line.

IMPLICIT INTEGER (A-Z)

Then, move the cursor to the last line of the range to be copied.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File PARTS  Lines 1 thru 3 Size 3

This file contains a part of another file copied directly from the
other file.  The following text is taken from file MYFILE:
```

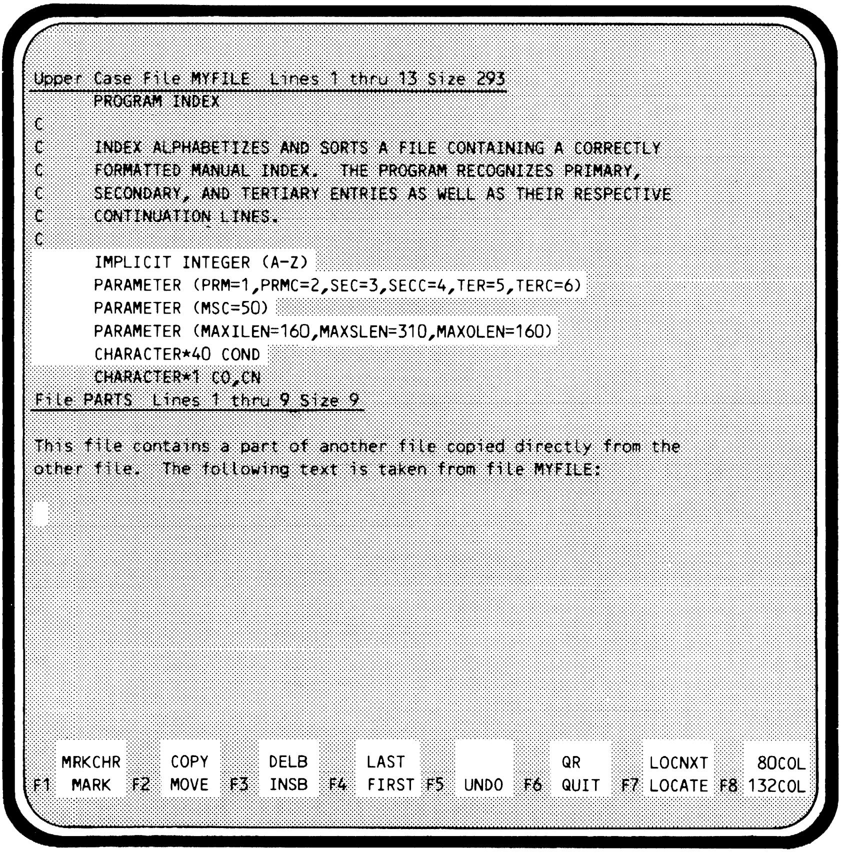
Press:

F1 **MARK**

again and FSE highlights the range of lines marked.

```
IMPLICIT INTEGER (A-Z)
PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
PARAMETER (MSC=50)
PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
CHARACTER*40 COND
```

Next, move the cursor to the point in file PARTS at which the lines from file MYFILE are to be inserted.

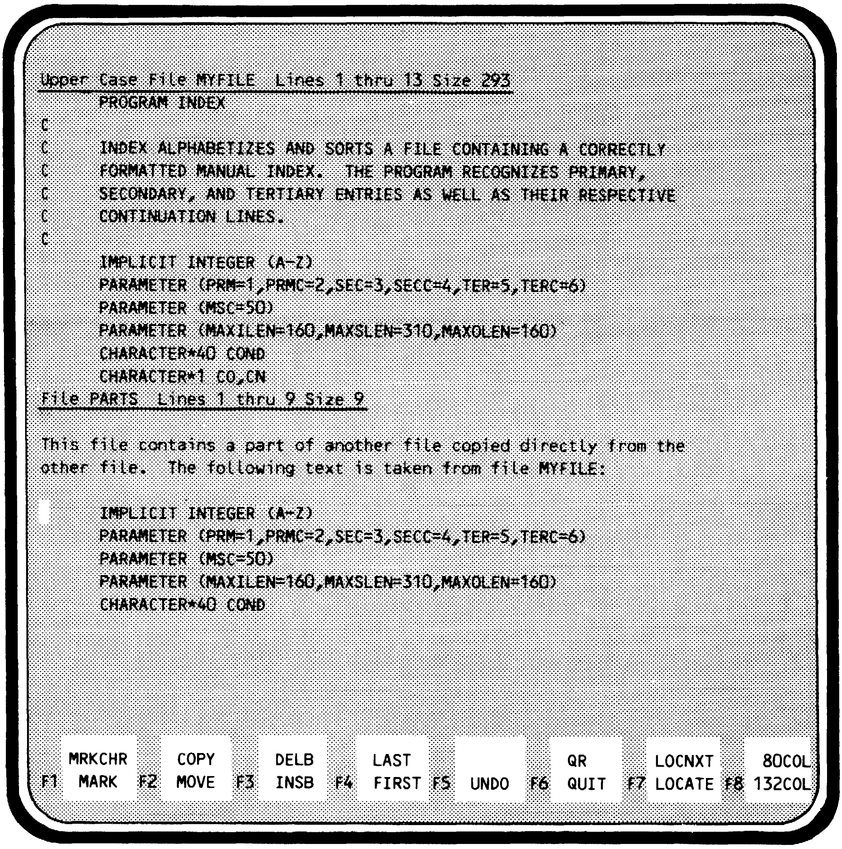


```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File PARTS  Lines 1 thru 9 Size 9

This file contains a part of another file copied directly from the
other file.  The following text is taken from file MYFILE:
```

Press:

⇧ F2 COPY

to insert the lines in file PARTS.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File PARTS  Lines 1 thru 9 Size 9

This file contains a part of another file copied directly from the
other file.  The following text is taken from file MYFILE:

      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
```

To return to editing file MYFILE only, press:

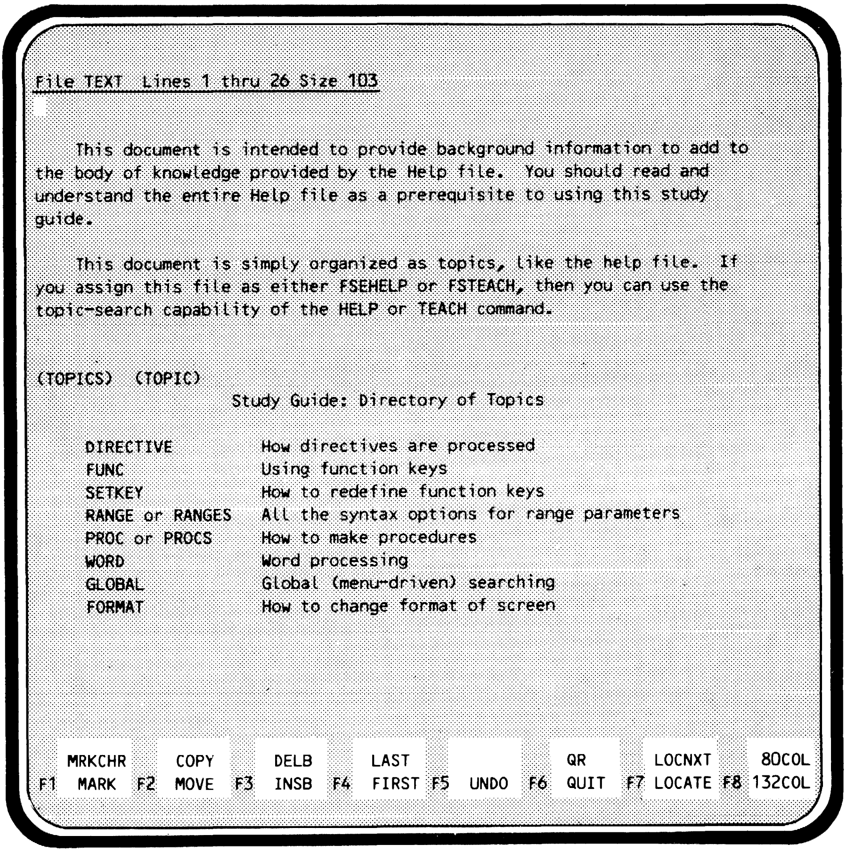
BACK

To edit a different permanent file (TEXT), created in the 6/12-bit ASCII character code set, you would press:

HOME

and enter:

```
F TEXT A G
```



```
File TEXT  Lines 1 thru 26 Size 103


    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.


(TOPICS)  (TOPIC)
                    Study Guide: Directory of Topics

     DIRECTIVE         How directives are processed
     FUNC              Using function keys
     SETKEY            How to redefine function keys
     RANGE or RANGES   All the syntax options for range parameters
     PROC or PROCS     How to make procedures
     WORD              Word processing
     GLOBAL            Global (menu-driven) searching
     FORMAT            How to change format of screen
```

Our editing example requires the use of the function key prompts that contain some of the word processing functions. To display the additional prompts, press:

(HOME)

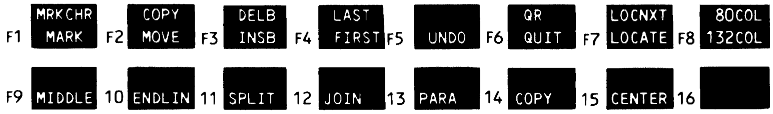
and enter:

```
S P 2
```

which stands for:

```
SET PROMPT 2
```

FSE displays all 16 function key prompts.



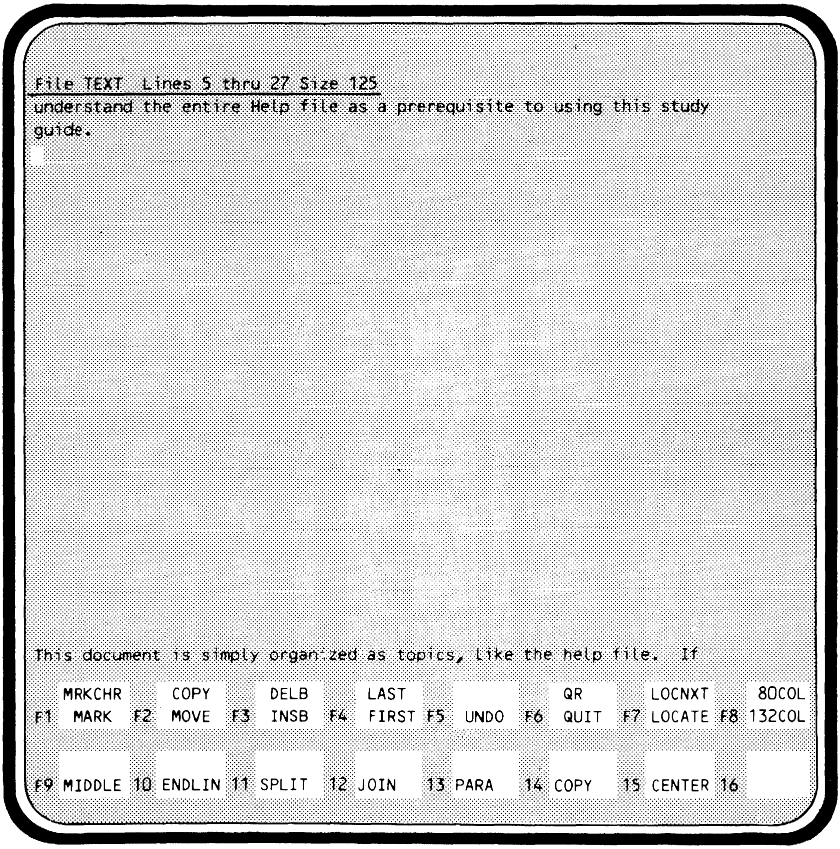
To insert a paragraph between the first and second paragraphs, position the cursor between them and press:

F3 **INSB**

(A paragraph is a group of lines that does not contain a blank line.)

```
    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.
```

FSE inserts blank lines over which you can type your new paragraph.



```
File TEXT  Lines 5 thru 27 Size 125
understand the entire Help file as a prerequisite to using this study
guide.




















This document is simply organized as topics, like the help file.  If
```

```
   MRKCHR      COPY      DELB      LAST                  QR        LOCNXT      80COL
F1 MARK    F2  MOVE  F3  INSB  F4  FIRST  F5  UNDO   F6  QUIT  F7  LOCATE  F8  132COL

F9 MIDDLE  10  ENDLIN 11 SPLIT 12  JOIN   13  PARA   14  COPY  15  CENTER  16
```

Then, type the new paragraph.

```
     To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.▒
```

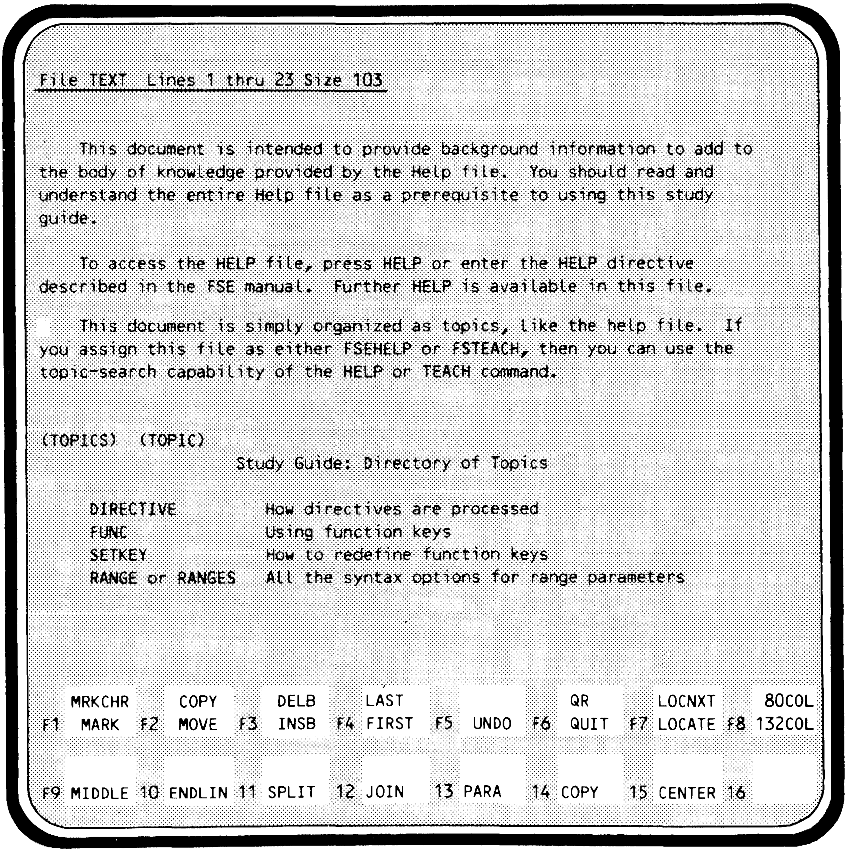
Position the cursor anywhere in the new paragraph and press:

F13 **PARA**

to ensure that the lines conform to the boundaries set by the SET WORD FILL directive (defaults are 1, 65, and 5).

To delete the unused blank lines, press:

⇧F3 **DELB**



```
File TEXT  Lines 1 thru 23 Size 103


     This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

     To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.

     This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the
topic-search capability of the HELP or TEACH command.


(TOPICS)  (TOPIC)
                         Study Guide: Directory of Topics

      DIRECTIVE           How directives are processed
      FUNC                Using function keys
      SETKEY              How to redefine function keys
      RANGE or RANGES     All the syntax options for range parameters



    MRKCHR     COPY       DELB      LAST                   QR        LOCNXT      80COL
F1  MARK   F2  MOVE   F3  INSB  F4  FIRST  F5  UNDO   F6  QUIT   F7  LOCATE  F8  132COL

F9 MIDDLE 10 ENDLIN 11 SPLIT  12 JOIN   13 PARA   14 COPY   15 CENTER 16
```

To enter a heading above the new paragraph, position the cursor at the first line of the paragraph.

```
  ▒ To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
```

Press:

(⇧) (INSRT)

twice. Then enter the heading.

```
Paragraph Number 2 ▒

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
```

To center this heading, position the cursor anywhere on the heading line.

```
Paragraph Number 2 ▒

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
```

Press:

F15 CENTER

```
▒                         Paragraph Number 2

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
```

To return the screen to displaying only the first line of programmable function key prompts, press:

(HOME)

and enter:

```
S P 1
```

Changing Your Screen's Format

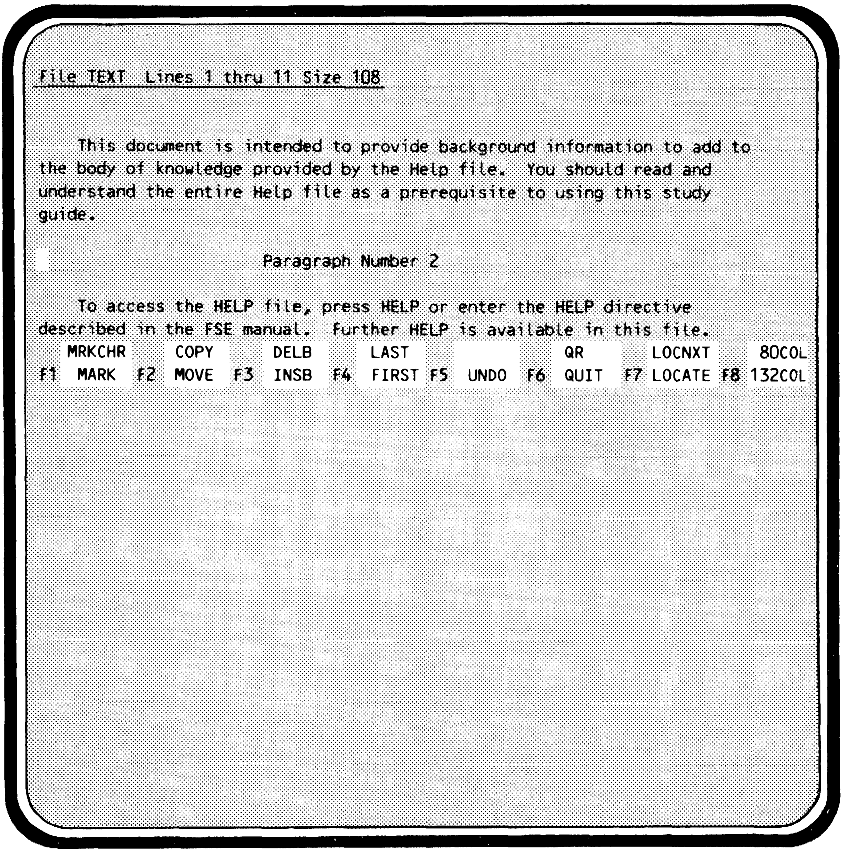
You might prefer to display a smaller number of lines on the screen than the default number. For example, to limit the screen to only 14 lines, including the file header and function key prompts, press:

HOME

and enter the SET VIEW LINE directive.

```
S V L 14
```

The first screen for file TEXT would then appear as:



```
File TEXT  Lines 1 thru 11 Size 108

    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

                        Paragraph Number 2

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
   MRKCHR     COPY      DELB      LAST                QR      LOCNXT     80COL
F1 MARK   F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT F7 LOCATE F8 132COL
```

The remaining space on the screen is left blank and is not used by FSE.

Stopping FSE

To stop FSE and make the changes to files TEXT and PARTS permanent, enter the QUIT REPLACE directive by pressing:

⇧F6 QR

The following messages appear.

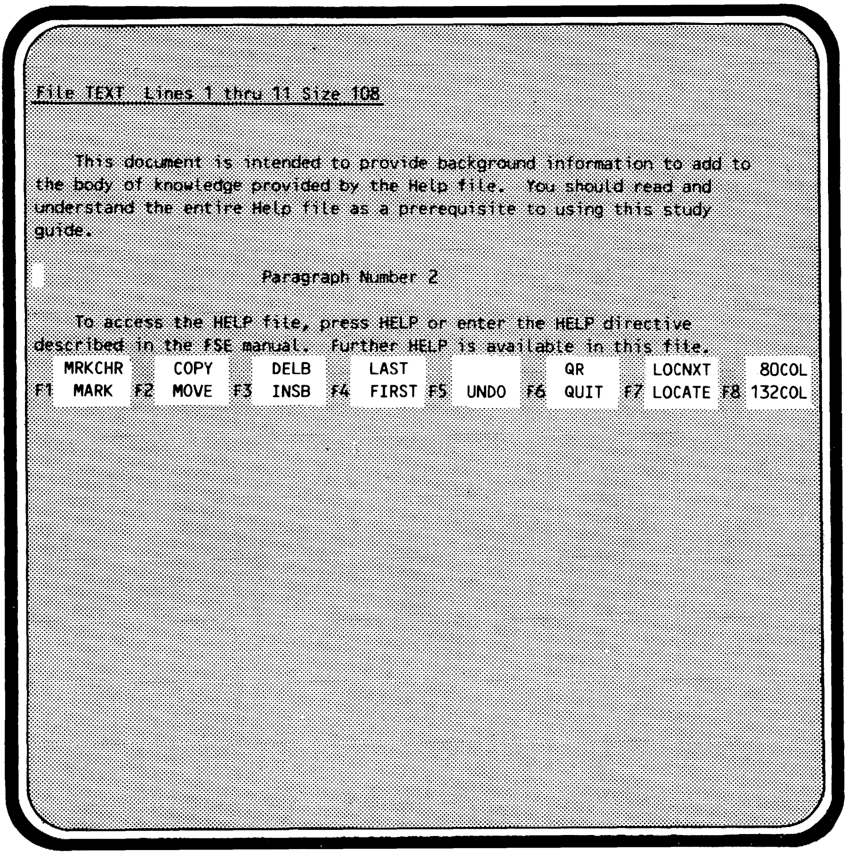
```
FILE:  MYFILE  (NO CHANGES)
FILE:  PARTS   (PERMANENT)
FILE:  TEXT    (PERMANENT)
FILE:  FSEPROC (NO CHANGES)
```

Restarting FSE

During a terminal session, you can return to editing a file at the exact spot you left it by restarting FSE. For example, to return to sample file TEXT, enter:

```
FSE
```

The screen appears just as you left it.



```
File TEXT  Lines 1 thru 11 Size 108


    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

                          Paragraph Number 2

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
   MRKCHR     COPY      DELB      LAST                QR      LOCNXT    80COL
F1  MARK   F2 MOVE   F3 INSB   F4 FIRST F5  UNDO   F6 QUIT  F7 LOCATE F8 132COL
```

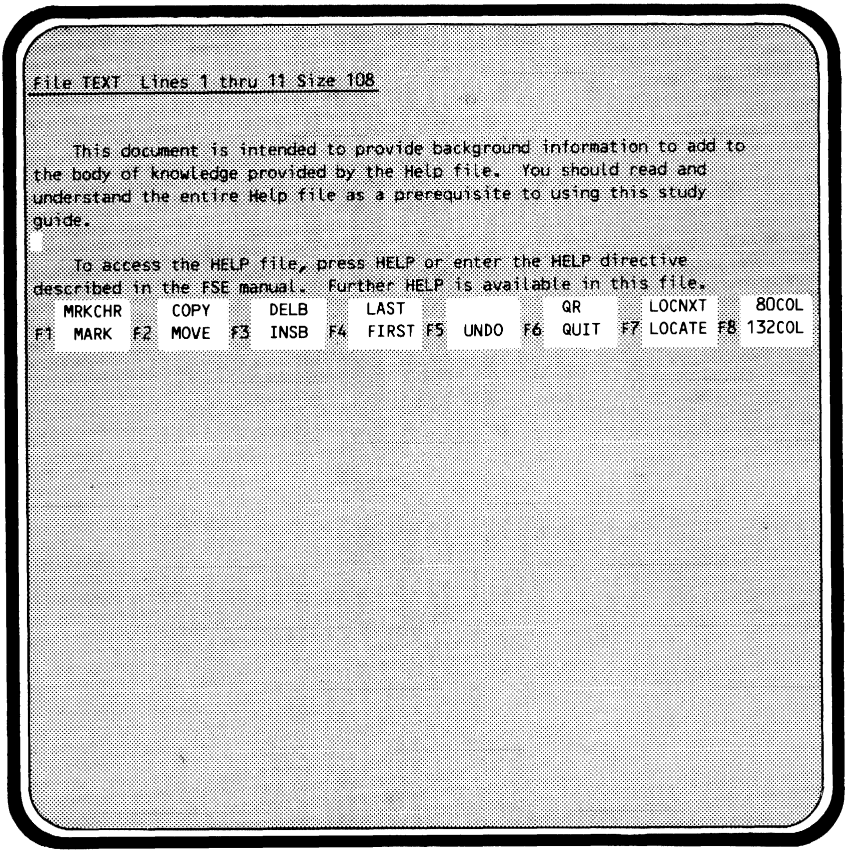
You can then continue editing. For example, to delete the paragraph heading, position the cursor on the heading line and press:

⇧ DLETE

Press:

⇧ DLETE

again to remove the extra blank line. The following screen results.



```
File TEXT  Lines 1 thru 11 Size 108

    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.
   MRKCHR     COPY      DELB      LAST                QR     LOCNXT     80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT F7 LOCATE F8 132COL
```

Enter:

```
QR
```

or press:

⇧ F6 QR

(if you have redefined it) to stop FSE and make the change to file TEXT permanent.

Advanced FSE Functions 6

Advanced FSE Functions 6

This section describes the following advanced features of FSE.

- Editing two files at the same time.
- Creating and using FSE procedures.

NOTE

You should be familiar with the directives before you attempt to create FSE procedures.

- Redefining programmable function keys.
- Labeling programmable function key prompts.
- Procedure branching.
- Changing your screen's format.

NOTE

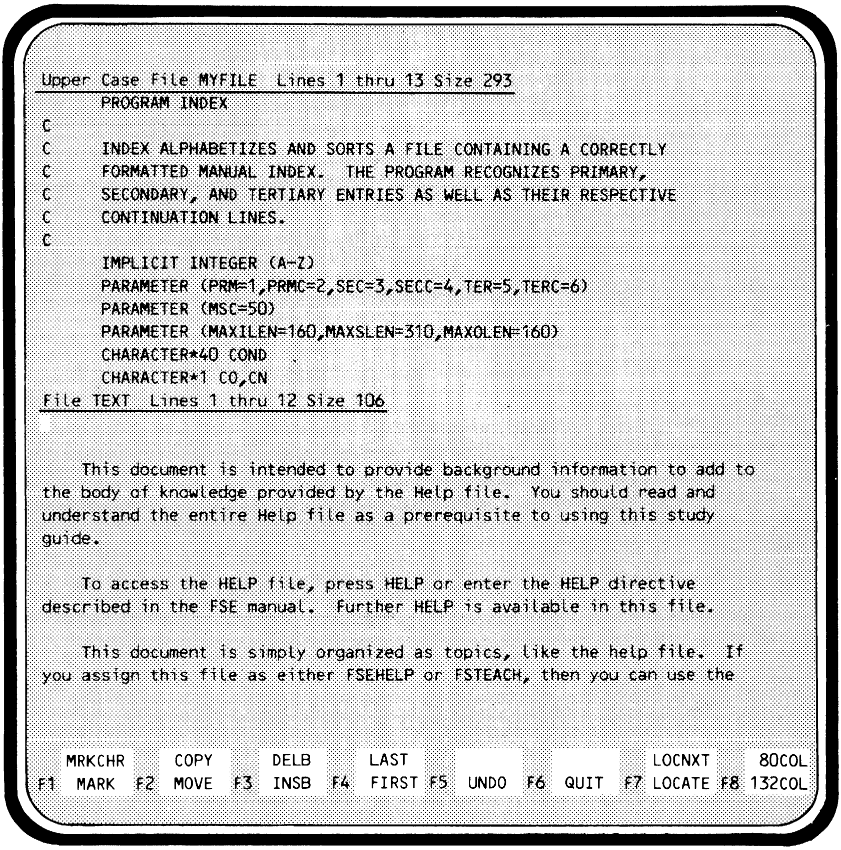
For each task, sample display screens will show you how an imaginary file would look before and after editing. The sample files are not released with the system. You must create your own sample file, if you want to perform the tasks in the examples.

Editing Two Files at the Same Time

In screen editing mode, FSE allows you to simultaneously display two different files or two different parts of one file. To do so, you must enter the FSE directive with the SPLIT parameter. For example, suppose you are editing file MYFILE, and you also want to edit permanent file TEXT in ASCII mode. Enter:

```
F TEXT A G S
```

FSE splits the screen, displaying file TEXT in the bottom half of the screen in CDC 6/12-bit ASCII code.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File TEXT  Lines 1 thru 12 Size 106


    This document is intended to provide background information to add to
the body of knowledge provided by the Help file.  You should read and
understand the entire Help file as a prerequisite to using this study
guide.

    To access the HELP file, press HELP or enter the HELP directive
described in the FSE manual.  Further HELP is available in this file.

    This document is simply organized as topics, like the help file.  If
you assign this file as either FSEHELP or FSTEACH, then you can use the


   MRKCHR     COPY      DELB      LAST                         LOCNXT     80COL
F1  MARK  F2  MOVE  F3  INSB  F4  FIRST F5  UNDO  F6  QUIT  F7 LOCATE F8 132COL
```

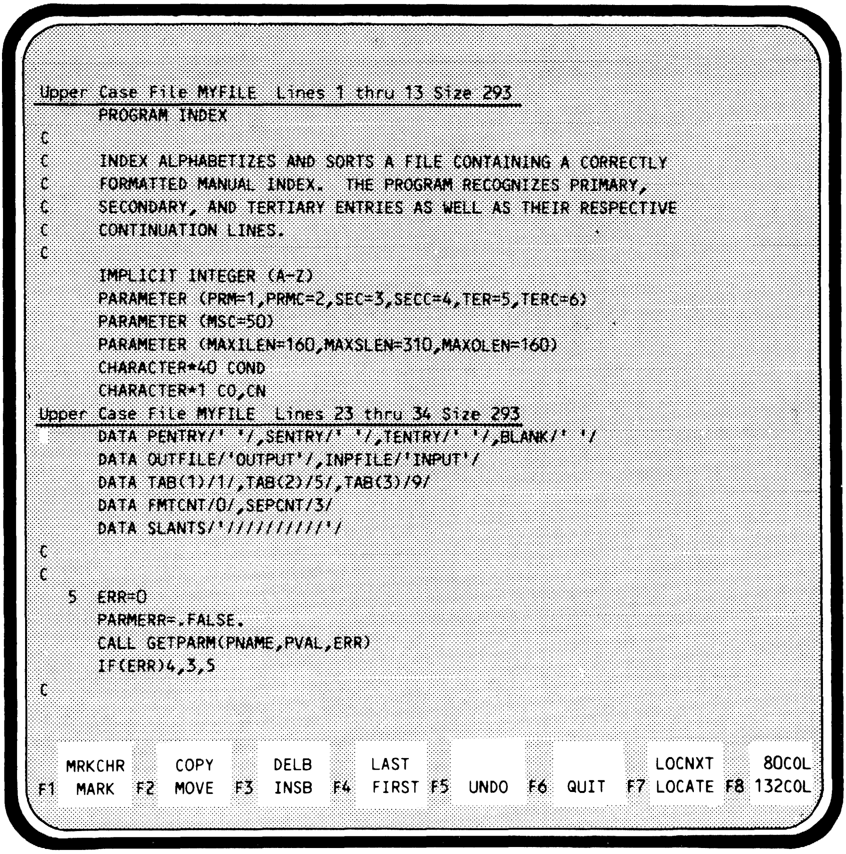
You can edit and page through the two files independently of each other. For example, if the cursor is positioned within the MYFILE text area and you press FWD, only file MYFILE is advanced one page.

To display different parts of the same file, specify the name of the current file on the FSE directive with the SPLIT parameter. For example, to display different parts of file MYFILE at the same time, enter:

```
F MYFILE S
```

FSE splits the screen, showing a copy of the first page of file MYFILE on the bottom half of the screen. Changes you make to one copy of the file are made simultaneously to the other.

You can page through each copy independently by positioning the cursor in the copy you want to edit. In the sample screen below, the first page of MYFILE is on the top half; another part of file MYFILE is on the bottom half.



```
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
Upper Case File MYFILE  Lines 23 thru 34 Size 293
      DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
      DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
      DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
      DATA FMTCNT/0/,SEPCNT/3/
      DATA SLANTS/'//////////'/
C
C
    5 ERR=0
      PARMERR=.FALSE.
      CALL GETPARM(PNAME,PVAL,ERR)
      IF(ERR)4,3,5
C
```

Creating and Using FSE Procedures

FSE enables you to create procedures containing any of the FSE directives described in section 4. Each procedure is usually contained on a separate record within a file named FSEPROC. If you do not have a local file or a permanent file by that name, FSE accesses a site-defined version of it and makes it local each time you start. You may alter this copy and save it to be used during editing.

The first line of each procedure must be the procedure's name (from one through seven characters). Procedures must be separated by either an end-of-record (EOR) statement or a QUIT PROC (QP) directive. For example, an FSEPROC file could contain the following procedures.

```
STARTUP
S K 5 S/LP/
(EOR)
PROC1
S K 1 S/LP/
QP
PROC2
LN;V
QP
```

NOTE

Because a record requires a minimum of one PRU, using the QP directive to separate procedures may save disk space.

To execute the procedures within FSEPROC, use the - (dash) directive. For example, you have the following procedure on file FSEPROC.

```
PROC1
S K 6 S/L N 2/
QP
```

To execute PROC1 enter:

```
-PROC1
```

FSE goes to file FSEPROC and looks for procedure PROC1. When PROC1 is executed, the shifted F6 key is redefined to execute the LOCATE NEXT 2 (L N 2) directive.

To call a procedure from a local or permanent file other than FSEPROC, include the other file name in parentheses. The following example executes procedure SEARCH from file ALTFILE.

```
-SEARCH (ALTFILE)
```

If you have certain functions you want executed every time you start FSE, use the procedure name STARTUP in your FSEPROC file. Each time you start (but not restart) FSE, it looks for the record named STARTUP on FSEPROC and executes the FSE directives.

For example, to redefine the shifted F5 key as the LOCATE PREVIOUS function, create the following procedure.

```
STARTUP
S K 5 S/ LP/
QP
```

Each time you start FSE, the STARTUP procedure redefines the shifted F5 key.

NOTE

To greatly increase the efficiency and speed of batch processing operations, move the procedures that will be used most often to the beginning of FSEPROC (or your own procedure file).

SET ANNOUNCE Directive

The SET ANNOUNCE directive enables you to specify, within a procedure, the prompts or messages to be displayed on the right side of the FSE directive line.

For example, suppose you have a procedure called TOP that appears as:

```
TOP
P 1
QP
```

This procedure positions you at the first line of a file. If you want the message:

```
FIRST LINE OF THE FILE
```

to be printed each time this procedure is executed, add the following line to the procedure.

```
S A/FIRST LINE OF THE FILE/
```

From now on, every time you use the TOP procedure to go to the first line of a file,

```
FIRST LINE OF THE FILE
```

is printed in the message area of the FSE directive line.

This message disappears when you perform the next function. The maximum message length is 40 characters. Refer to section 4 for more information on the SET directive.

Micros

Micros enable you to specify the current character, word, line, or file as a parameter either in an FSE procedure or in a directive string assigned to a programmable function key. You can also create your own prompts for input or specify different parameters each time you call a procedure. These micros, all preceded by an ampersand (&), are:

Micro	Function
&C	Specifies the current column number.
&F	Specifies the current file name.
&L	Specifies the current line number.
&T	Specifies the current terminal type (Viking 721, CDC 722, VT100, and so forth).
&W	Specifies the word on which the cursor is positioned, or, if the cursor is not positioned on a word, the next word to the right.
&n	Specifies the nth parameter of a procedure call.
&?	Enables you to enter procedure parameters interactively.
&&	Specifies an ampersand (&).

Specifying the Current Column Number (&C)

To specify the current column number, use the &C micro. For example, the following procedure uses the current column number as the parameter in a SET VIEW OFFSET directive.

```
COL
S V O &C
QP
```

When you enter:

```
-COL
```

the current column number becomes the left margin on the screen.

Specifying the Current File (&F)

To specify the current file, use the &F micro. For example, the following procedure compiles the current file as a FORTRAN 5 program, no matter what the current file is.

```
FTN
/FTN5,I=&F,L=0.
QP
```

Each time you enter:

```
-FTN
```

the current file is compiled as a FORTRAN 5 program. To return to the point in the file where you entered -FTN, enter:

```
FSE
```

Specifying the Current Line (&L)

To specify the current line of a file, use the &L micro. For example, the following procedure copies the current line to end of the file.

```
CPYL
COPY&L TO LAST
QP
```

When you enter:

```
-CPYL
```

the current line is copied to the end of the file.

Specifying the Current Terminal Type (&T)

To create a procedure whose execution depends on the type of terminal being used, use the &T micro. For example, using this micro you can create the following STARTUP procedure, which calls a second procedure based on the current terminal type.

```
STARTUP
-XX&T
QP
XX721
S K S 6/GET STATUS/
S A/STATUS IS ON SHIFT-F6/
QP
XX722
S K S 9/GET STATUS/
S A/STATUS IS ON SHIFT-F9/
QP
```

The STARTUP procedure automatically changes the keyboard functions, depending on whether a Viking 721 or a CDC 722 terminal is being used.

Specifying the Current Word (&W)

To specify the current word of a file, use the &W micro. For example, the following procedure locates the next occurrence of the current word and centers vertically on the screen the line containing it.

```
NW
L N/&W/;V
QP
```

If the cursor is positioned at the word INTEGER and you enter:

```
-NW
```

FSE locates the next occurrence of INTEGER and centers the line in which it appears.

Specifying the nth Parameter of a Procedure Call (&n)

To specify different parameters for a procedure each time the procedure call is used, use the &n micro (n is the number of the parameter you want to specify). For example, the following procedure is on FSEPROC.

```
TOEND
C &1 &2 TO L
QP
```

This procedure copies the range you specify to the end of the file. Using this procedure, you can copy lines 20 through 30 to the end of the file by entering:

```
-TOEND 20 30
```

The next time you can specify a different range by including it on the call to the TOEND procedure.

Entering Parameters Interactively (&?)

To create a procedure that prompts you for input, use the &? micro. For example, the following FIND procedure prompts you to enter the string you want to find and then centers the line containing the string.

```
FIND
L/&?/;V
QP
```

When you enter:

```
-FIND
```

the prompt:

```
<--- ENTER TEXT
```

appears in the FSE message area. If you then enter DATA, FSE locates the next occurrence of DATA and centers the line containing it on the screen.

`ENTER TEXT` is the default prompt. To create your own prompts, include a SET ANNOUNCE directive in your procedure before the directive containing the &? micro. For example, to create the prompt `FIND WHAT?` in the FIND procedure, insert the SET ANNOUNCE directive as follows:

```
FIND
S A/FIND WHAT?/
L/&?/;V
QP
```

Now when you enter:

```
-FIND
```

you are prompted with `FIND WHAT?` instead of `ENTER TEXT`.

Micro Values

You may write a procedure in which the value of a micro changes within the procedure, as for example in:

```
PROC 1
.P3
.P&C+4
.P&C+7
.P&C+3
QP
```

FSE changes the value of the &C micro (which represents the current column) at the end of each line of the procedure and, therefore, positions itself sequentially at columns 3, 7, 14, and 17.

The procedure just described has only one directive per line. Suppose that you wrote a procedure using the same directives, but combined two of the directives onto one line.

```
PROC 1A
.P3
.P&C+4;.P&C+7
.P&C+3
QP
```

Although the directives are identical, the results of the two procedures are different: PROC 1A positions FSE at columns 3, 10, and 13. Part of the reason for this difference is that FSE interprets all the micros on a line before it processes the directives. As a result, each occurrence of a particular micro on a line is interpreted the same. In PROC 1A, FSE positions itself at column 3 at the end of the first line. It then interprets both of the micros in the second line as 3.

The different results occur for another reason. When more than one directive on a line is associated with the same micro, FSE uses the final directive as the determining value. In PROC1A, it uses the last directive on the line and positions itself at column 10.

Procedure Looping

Within FSE, you can create a procedure that repeats itself as long as certain conditions are met. This is called procedure looping.

Following are two versions of an identical procedure, both of which create loops, but only one of which functions correctly.

Version A	Version B

`PROC 1`
`.P&L+1;R/abc/xyz/;-PROC 1`
`QP`
	`PROC 1`
	`.P&L+1`
	`R/abc/xyz/`
	`-PROC 1`
	`QP`

In both versions of this procedure, the position of the current line will be increased by 1; abc will be replaced with xyz; and the recursive procedure call (a procedure call to itself) will cause the same procedure to be executed again.

The versions differ in that version A of the procedure provides for an eventual exit from the procedure, while version B causes an infinite loop.

The reason lies in the way FSE executes directives that occur on the same line of a procedure. If it is unable to execute a directive, it ignores all the following directives on that line, and immediately proceeds to the next line. In version A, if it cannot execute either the position or replace directive, it will skip the recursive call and execute the quit procedure directive.

In version B, if it is unable to execute these directives, it will proceed to the recursive call and never reach the quit procedure directive.

The general rule to remember is that if you have a recursive call to a procedure, place the exit condition on the same line as the recursive call.

Conditional Procedures

The execution of conditional procedures depends on the occurrence of a specific condition or event (for example, the end of the file, x being greater than y, or the location of specific characters). The following examples illustrate the three methods available for writing conditional procedures.

Example 1

You can use the LOCATE or REPLACE directive.

Procedure	Description
`FNNDEL`	Procedure name.

`L/go away/;D;PN;-FNNDEL`
`PF`
`(EOR)`
	Delete all lines containing the phrase *go away*, beginning with the current one. After all the lines have been deleted, return to the first line.

Example 2

You can combine directives and line addresses.

Procedure	Description
`DELALT`	Procedure name.

`D;PN;PN;-DELALT`
`(EOR)`
	Delete the current line and advance one line if possible. Advance a second line if possible. If both line advances have occurred, repeat the procedure.

Example 3

You can use the QUIT PROC directive to explicitly terminate a procedure. This can be accomplished using either no, one, or two parameters.

Using No Parameters

Procedure	Description
`DELCUR`	Procedure name.

`LCC/don't shoot/;QP`
`D`
`QP`
	Search for the phrase *don't shoot* on the current line. If successful, go to QP and quit. If not successful, delete the line and terminate the procedure.

Using One Parameter

The address C+10 is the parameter for this procedure. The procedure will only execute as long as there are at least 10 lines remaining after the current line.

Procedure	Description
`DELTEN`	Procedure name.

`QP C+10`
`D`
`(EOR)`
	If there are 10 or more lines after the current line, delete the current line. Otherwise, terminate the procedure.

Using Two Parameters

Here there are two parameters: the value for x and the value for y (which is the incremented value of x). The file used must be either ASCII or ASCII 8.

Procedure	Description
UPCASE	Procedure name.
QPx y R/a/A/xx R/b/B/xx R/c/C/xx . . . R/z/Z/xx	If x is less than or equal to y, capitalize all the letters in x.
Sx x+1;-UPCASE (EOR)	If x is still in the file, add 1 to x and repeat the procedure. Otherwise, end the procedure.

Recovering from Errors in FSE Procedures

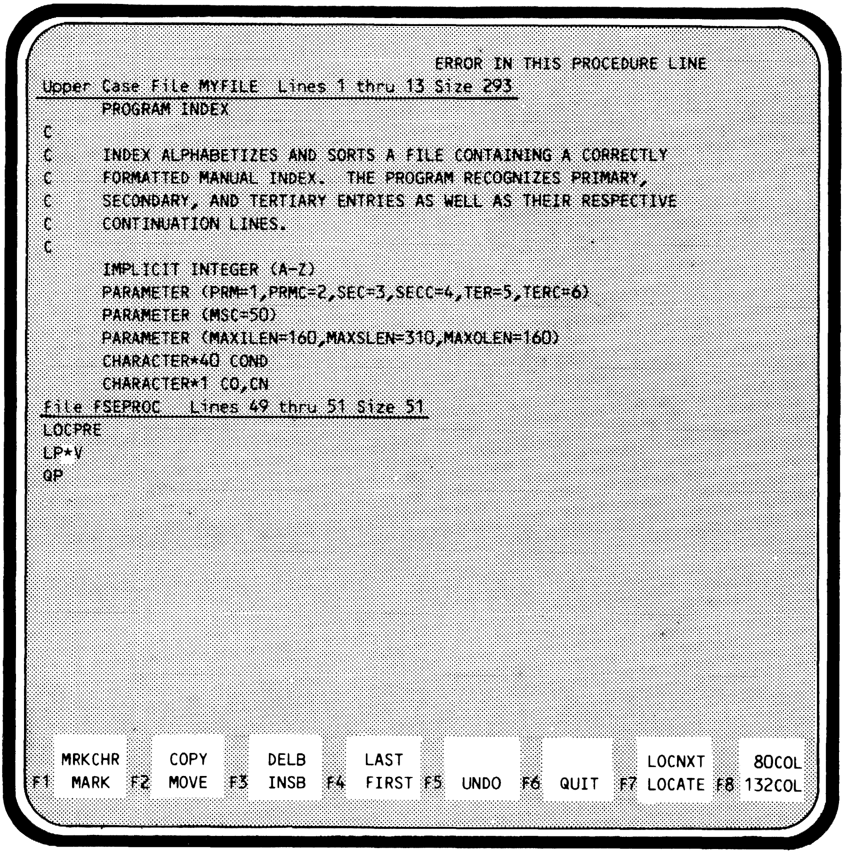
If you run a procedure containing an error, FSE accesses the file containing the procedure (FSEPROC or any other file) and positions you at the error. For example, assume FSEPROC contains the following procedure.

```
LOCPRE
LP*V
QP
```

The * is not a valid directive separator and will cause an error in the LOCPRE procedure. Suppose that while editing file MYFILE, you want to locate the previous occurrence of a string and center it on the screen. You enter:

```
-LOCPRE
```

FSE reads procedure LOCPRE and determines that it contains an error. FSE then displays file FSEPROC on the lower half of a split screen and positions the cursor at the error.



```
                                        ERROR IN THIS PROCEDURE LINE
Upper Case File MYFILE  Lines 1 thru 13 Size 293
      PROGRAM INDEX
C
C     INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
C     FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
C     SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
C     CONTINUATION LINES.
C
      IMPLICIT INTEGER (A-Z)
      PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
      PARAMETER (MSC=50)
      PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
      CHARACTER*40 COND
      CHARACTER*1 CO,CN
File FSEPROC   Lines 49 thru 51 Size 51
LOCPRE
LP*V
QP
```

You can then correct the error and execute the procedure again.

If you already have a split screen, the file on the lower half of the screen is replaced with the file containing the procedure error. To return to your previous edit file, either press:

(BACK)

or enter the BACK directive.

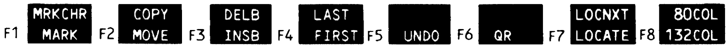
Redefining Programmable Function Keys

You can redefine the programmable function keys (the default functions are described in section 2) to execute any FSE directive. The redefined functions remain in effect for the duration of the current editing session.

To redefine a key, use the SET directive with the KEY parameter. For example, to redefine the F6 QUIT key as the QUIT REPLACE function, enter:

```
S K 6/ QR/
```

FSE then displays the new function key prompts.



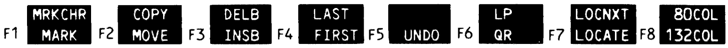
NOTE

Function key definitions must be 244 or fewer characters (only the first 6 characters will appear on the function key prompt).

To define or redefine the shifted function keys, include the SHIFT parameter. For example, to define the shifted F6 key as the LOCATE PREVIOUS function, enter:

```
S K 6 S/ LP/
```

FSE then displays the new function key prompts.



To return a key to its default function, enter the SET KEY directive with no other parameters. For example, to return the unshifted F6 key to QUIT, enter:

```
S K 6
```

If you want the redefined function keys to remain in effect from one editing session to the next, include a SET KEY directive within the STARTUP procedure on your FSEPROC file. Refer to *Creating and Using FSE Procedures* earlier in this section for information on the FSEPROC file.

Using FSE procedures, it is possible to redefine keys to call procedures. For example, you may want to redefine several keys on the first row of function key prompts to perform word processing functions. A simple way to do this is to create a procedure that redefines the keys you want to use. The following procedure illustrates this.

```
WORD
S K 1/ .FILL/
S K 3/ I W/
S K S 3/ D W/
S K S 6/ -DEF/
S K 8/ .JOIN/
S K S 8/.SPLIT/
QP
```

Spaces are included in the directive strings to make the function key prompts easier to read.

You then add to your STARTUP procedure the following line, defining the shifted F6 key as -WORD.

```
S K 6 S/ -WORD/
```

From then on, every time you press:

```
⇧ F6 -WORD
```

the function keys are redefined to the specified word processing functions.

Labeling Programmable Function Key Prompts

If you do not indicate otherwise, the prompt FSE displays for a programmable function key is the same as the directive string you entered when redefining the key. If, for example, you defined the shifted F6 key as:

```
S K 6 S/ LP;V/
```

the F6 function key prompt is:

```
   LP;V
F6 QUIT
```

You can, however, label the key prompt differently by using the LABEL parameter with the SET KEY directive. You could, for example, redefine a key to execute two directives, as just described, but only include one function on the key prompt. In the preceding example, you could label the key prompt LOCPRE, without affecting the two directives it executes, by entering:

```
S K S 6/ LP;V/L/LOCPRE/
```

The F6 prompt is:

```
   LOCPRE
F6  QUIT
```

but both the LOCATE PREVIOUS and VIEW directives are executed when you press the shifted F6 key.

NOTE

Function key labels must be six or fewer characters to fit in the prompt area at the bottom of the screen.

Labeling keys with descriptive phrases rather than the actual directive strings themselves helps you better understand each key's function and presents a cleaner looking screen.

If you want to change the label of a key but leave the directive string associated with the key unchanged, include only the LABEL parameter and the new label. For example, to change the label of the shifted F6 key from LOCPRE to LCTPRV, enter:

```
S K S 6 L/LCTPRV
```

The shifted F6 key is now labeled LCTPRV while the directive string associated with it (LP;V) remains the same.

Procedure Branching

In describing the way to redefine a programmable function key to call a procedure (*Redefining Programmable Function Keys* in this section), we created procedure WORD as follows:

```
WORD
S K 1/ .FILL/
S K 3/ I W/
S K S 3/ D W/
S K S 6/ -DEF/
S K 8/ .JOIN/
S K S 8/.SPLIT/
QP
```

The shifted F6 key was redefined to call this procedure (-WORD). Notice that procedure WORD again redefines the shifted F6 key, this time to call another procedure, procedure DEF. No limits are placed on the number of times you can branch off a default setting for a key, creating a procedure that redefines a key to call another procedure that redefines a key and so forth. To return to the original settings for the keys, create a procedure on FSEPROC that sets the function keys to their default. In example procedure WORD, this is the function of procedure DEF.

If the number of procedure branches you create is large, you might find it difficult to keep track of which procedure defines the keys in which way. To help you remember how your procedures redefine keys, a worksheet is provided on the next page. The following example shows how you would use the worksheet if you created the sample procedure WORD.

ORIGINAL

F1	F2	F3	F4	F5	F6	F7	F8
MRKCHR	COPY	DELB	LAST		-WORD	LOCNXT	80COL
MARK	MOVE	INSB	FIRST	UNDO	QUIT	LOCATE	132COL

F9	F10	F11	F12	F13	F14	F15	F16
MIDDLE	ENDLIN	SPLIT	JOIN	PARA	COPY	CENTER	

WORD

F1	F2	F3	F4	F5	F6	F7	F8
MRKCHR	COPY	D W	LAST		-DEF	LOCNXT	.SPLIT
.FILL	MOVE	I W	FIRST	UNDO	QUIT	LOCATE	.JOIN

F9	F10	F11	F12	F13	F14	F15	F16
MIDDLE	ENDLIN	SPLIT	JOIN	PARA	COPY	CENTER	

F1	F2	F3	F4	F5	F6	F7	F8

F9	F10	F11	F12	F13	F14	F15	F16

F1	F2	F3	F4	F5	F6	F7	F8

F9	F10	F11	F12	F13	F14	F15	F16

F1	F2	F3	F4	F5	F6	F7	F8

F9	F10	F11	F12	F13	F14	F15	F16

F1	F2	F3	F4	F5	F6	F7	F8

F9	F10	F11	F12	F13	F14	F15	F16

Changing Your Screen's Format

The type of screen format you will want to use depends on the operation you intend to perform. If you need to change the format of your screen, use the SET VIEW directive and various keyword parameters. With the SET VIEW directive you can:

- Specify the number of lines displayed each time the screen is rewritten.
- Specify the number of columns to be displayed each time the screen is rewritten.
- Specify the number of lines available for the lower half of a split-screen edit.
- Define the number of columns that can be edited.
- Change the line length limit.
- Change the first column displayed at the left edge of the screen.

Specifying the Number of Lines Displayed

If you do not want 26 lines (the default value) printed each time the screen is rewritten, use the SET VIEW LINE directive. For example, to have only 15 lines displayed, enter:

```
S V L 15
```

During subsequent file editing only 15 lines are displayed each time the screen is rewritten. These 15 lines include the file header and programmable function key prompts.

Specifying the Number of Columns Displayed

To specify the number of columns you want displayed each time the screen is rewritten, use the SET VIEW COLUMN directive. For example, to print only columns 1 through 60 (the default is 80), enter:

```
S V C 60
```

Only characters in columns 1 through 60 will be displayed when the screen is rewritten. The minimum value allowed is 10. A value exceeding 80 changes the screen display to 132-column format (if your terminal has this capability).

Specifying the Number of Lines Used for Split-Screen Editing

When split-screen editing, you can reduce the number of lines reserved for the lower half of the screen. To do so, use the SET VIEW SPLIT directive as shown in the following example.

```
S V S 7
```

Subsequent split-screen editing will use only seven lines for the bottom half of the screen.

Specifying Columns That Can Be Edited

To limit the number of columns that can be edited, use the SET VIEW EDIT directive. For example, if you want to edit only the first seven columns of a file, enter:

```
S V E 7
```

Subsequent editing will affect only columns 1 through 7. The left margin is always column 1.

Changing the Line Length Limit

To change the line length limit, enter the SET VIEW WARN directive. For example, to change the line length to 70 from a previous setting, enter:

```
S V W 70
```

When a line exceeds 70 characters, FSE will stop directive processing and issue a warning message.

Changing the First Column Displayed

To view lines that run off the edge of your screen, use the SET VIEW OFFSET directive to specify the leftmost column to be displayed. For example, to view a line beginning at column 30, enter:

```
S V O 30
```

Column 30 then becomes the leftmost column displayed on your screen. To return to the default setting (1), enter:

```
S V O 1
```

Column 1 is once more the leftmost column displayed.

Line Editing 7

FSE is not exclusively a screen editor; it also functions as a line editor, which can be used on any terminal. All but a few of the directives available for screen editing are also applicable to line editing. When line editing, enter directives after the ?? prompt and then always press:

(NEXT)

More information on the directives is provided in section 4, *FSE Directives*.

NOTE

For each task, examples will show you how an imaginary file would look before and after editing. The sample files are not released with the system. You must create your own sample files if you want to perform the tasks in the examples.

Getting Started

Setting Line Mode

To use FSE as a line editor, you must be logged in to NOS. For information on NOS login procedures, refer to the *NOS 2 Reference Set, Volume 2, Guide to System Usage*.

Once logged in to NOS, enter the NOS LINE command.

```
LINE
```

This sets your terminal in line mode or cancels a SCREEN command if you have been screen editing. If you have not been screen editing during your current terminal session, you do not need to enter the LINE command; FSE is a line editor by default.

Starting FSE

To start FSE, enter the following command, including the name of the file you want to edit (MYFILE is the sample file).

```
FSE,MYFILE
```

The following prompt appears.

```
NOS FULL SCREEN EDITOR
EDIT:  MYFILE
??
```

Creating a File

To create a file, enter the directive:

```
?? INSERT
```

to access the insert function (INSERT can be abbreviated to I). Line numbers are automatically generated as you enter text and press NEXT.

```
?? I
1 ? PROGRAM INDEX
```

Each use of NEXT advances you one line. Pressing NEXT twice returns you to the ?? prompt so that you can enter new directives.

```
  1 ? PROGRAM INDEX
  2 ? C
  3 ? INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
  4 ?
??
```

Typing `INSERT` once again returns you to the line you last called with a directive.

Editing

Displaying Lines

To display all the lines in a file, enter the PRINT ALL directive:

```
?? PA
```

To display lines 1 through 30 of the file, enter the following PRINT directive:

```
?? P1 30
```

The first 30 lines of the file are displayed.

```
 1          PROGRAM INDEX
 2 C
 3 C        INDEX ALPHABETIZES AND SORTS A FILE CONTAINING A CORRECTLY
 4 C        FORMATTED MANUAL INDEX.  THE PROGRAM RECOGNIZES PRIMARY,
 5 C        SECONDARY, AND TERTIARY ENTRIES AS WELL AS THEIR RESPECTIVE
 6 C        CONTINUATION LINES.
 7 C
 8          IMPLICIT INTEGER (A-Z)
 9          PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
10          PARAMETER (MSC=50)
11          PARAMETER (MAXILEN=160,MAXSLEN=310,MAX%%EN=-60)))..../
12          CHARACTER*40 COND
13          CHARACTER*1 CO,CN
14          CHARACTER*10 SLANTS
15          CHARACTER*12 FMTST
16          CHARACTER*(MAXILEN) INPLIN
17          CHARACTER*7 PVAL,PNAME
18          CHARACTER*7 INPFILE,OUTFILE
19          CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
20          LOGICAL PARMERR
21          DIMENSION TAB(3)
22 C
23          DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
24          DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
25          DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
26          DATA FMTCNT/0/,SEPCNT/3/
27          DATA SLANTS/'//////////'/
28 C
29 C
30       5  ERR=0
```

A directive that attempts to display more lines than exist in a file results in the message:

```
OUT OF BOUNDS
```

NOTE

The line numbers are added by FSE so you can see where you are in the file; they are not part of the file text. Also, this file, as shown here, contains a few errors which will be corrected as we proceed through this section.

Searching

Suppose you want to move the first line in the file containing the string:

```
LOGICAL PARMERR
```

to the line between present lines 16 and 17.

To locate the line, you must position the cursor at the first line of the file by entering the following PRINT directive.

```
?? P1
```

The first line is printed.

```
1       PROGRAM INDEX
```

Then, enter:

```
?? L/LOGICAL PARMERR
```

The first line containing the string is displayed.

```
20      LOGICAL PARMERR
```

To see a few lines before and after line 20, enter the following VIEW directive.

```
?? V
```

The four preceding lines, line 20, and the four following lines are displayed.

```
16         CHARACTER*(MAXILEN) INPLIN
17         CHARACTER*7 PVAL,PNAME
18         CHARACTER*7 INPFILE,OUTFILE
19         CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
20         LOGICAL PARMERR
21         DIMENSION TAB(3)
22 C
23         DATA PENTRY/' '/,SENTRY/' '/,TENTRY/' '/,BLANK/' '/
24         DATA OUTFILE/'OUTPUT'/,INPFILE/'INPUT'/
```

Inserting Lines

To insert:

```
   LOGICAL PARMERR
```

between lines 16 and 17, enter the following INSERT directive.

```
   ?? I16
```

FSE prompts you with:

```
   17 ?
```

You then enter the line to be inserted.

```
   17 ?\LOGICAL PARMERR
   18 ?
```

The \ is the default tab character. Because the default tabs are set at 7 and 72, `LOGICAL` starts at column 7.

You can enter as many lines as you want on any insertion, ending the insertion either by entering a tab character at the end of an inserted line or by pressing:

```
   (NEXT)
```

on an empty line.

You then display the surrounding lines with the VIEW directive.

```
   ?? V
   13        CHARACTER*1 CO,CN
   14        CHARACTER*10 SLANTS
   15        CHARACTER*12 FMTST
   16        CHARACTER*(MAXILEN) INPLIN
   17        LOGICAL PARMERR
   18        CHARACTER*7 PVAL,PNAME
   19        CHARACTER*7 INPFILE,OUTFILE
   20        CHARACTER*50 PENTRY,SENTRY,TENTRY,BLANK
   21        LOGICAL PARMERR
```

Notice that the line numbers following 17 have all increased by one to accommodate the new line.

NOTE

To insert characters while line editing, use the ALTER directive, described in section 4.

Deleting Lines

To delete LOGICAL PARMERR on line 21, enter:

```
?? D21
```

The deleted line is displayed at your terminal.

```
21        LOGICAL PARMERR
```

You can delete more than one line at a time. To delete lines 25 through 27, for example, enter the following DELETE directive.

```
?? D25 27
```

The deleted lines are displayed, and line numbers on the remaining text shift accordingly.

```
25         DATA TAB(1)/1/,TAB(2)/5/,TAB(3)/9/
26         DATA FMTCNT/0/,SEPCNT/3/
27         DATA SLANTS/'//////////'/
```

To avoid the display of these lines, include the QUIET parameter.

```
D 25 27 Q
```

(If you decide that you want these lines returned, use the UNDO directive.)

Getting Online Help

Suppose you then need to change line 11, but cannot quite recall how the ALTER directive works. To get help, enter the following directive.

```
?? H ALTER
```

This directive accesses the FSE HELP file and positions the cursor at the area of text describing the ALTER directive. To continue reading the HELP file text, enter PRINT directives as needed. For example, to read the next 20 lines, enter PN20. When you have read the needed information, the following directive returns you to file MYFILE.

```
?? BACK
```

NOTE

To delete characters while line editing, use the ALTER directive, described in section 4.

Changing Lines

You then enter the following PRINT directive to position the cursor to line 11.

```
?? P11
```

The line is displayed.

```
11        PARAMETER (MAXILEN=160,MAXSLEN=310,MAX%%EN=-60)))..../
```

You enter the following ALTER directive.

```
?? A
```

FSE responds with the following prompt.

```
 11       PARAMETER (MAXILEN=160,MAXSLEN=310,MAX%%EN=-60)))..../
A??
```

You then space over to where the change is to be made and enter the correction underneath the error.

```
 11       PARAMETER (MAXILEN=160,MAXSLEN=310,MAX%%EN=-60)))..../
A??                                             0L   1   !
```

The entries in this example perform the following functions.

Entry	Function
Space	No effect on the line.
0	Replaces first %.
L	Replaces second %.
1	Replaces -.
!	Truncates the line at the current character.

When you press (NEXT), the corrected line is displayed.

```
11        PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
```

Copying Lines

To copy lines 8 through 12 immediately after line 15, enter the following COPY directive:

```
?? C 8 12 TO 15
```

The copied lines and their new line numbers are then displayed.

```
16       IMPLICIT INTEGER (A-Z)
17       PARAMETER (PRM=1,PRMC=2,SEC=3,SECC=4,TER=5,TERC=6)
18       PARAMETER (MSC=50)
19       PARAMETER (MAXILEN=160,MAXSLEN=310,MAXOLEN=160)
20       CHARACTER*40 COND
```

Replacing Text

To replace all occurrences of INTEGER with INT, enter the following REPLACE directive.

```
?? RA/INTEGER/INT
```

The lines affected by the change are displayed.

```
8   IMPLICIT INT (A-Z)
71  INT(A1)
90  INT(B23)
```

After the replacement is completed, the cursor is positioned at the last line affected. In this example, the cursor would rest at line 90.

Undoing Changes

If you need to undo a change to your file, enter the UNDO directive. For example, to undo the last REPLACE directive, enter:

```
?? UNDO
```

All the INTEGERs that were replaced with INT are now changed back to INTEGER. To undo other changes made to your file during your current editing session, enter an UNDO directive for each item you want restored. The changes are undone in the reverse order of their occurrence. UNDO will not display the reinstated line; however, you can enter the PRINT directive to verify that the changes have been undone.

Stopping FSE and Returning to NOS

To stop FSE and return to NOS, enter:

```
?? QUIT
```

The following message then appears.

```
FILE:  MYFILE
```

This message tells you that you have made changes to file MYFILE but they are not permanent.

Restarting FSE

To return to FSE at the point you left, enter the FSE command without parameters.

```
FSE
```

In this example, you are positioned at line 90, the point at which you stopped FSE with the QUIT directive.

Stopping FSE and Making Your Changes Permanent

To stop FSE and make the changes to your file permanent, enter the following QUIT REPLACE directive.

```
QR
```

The following message appears, confirming that the changes have been made permanent.

```
FILE:  MYFILE (PERMANENT)
```

Editing Sequenced Files

When editing or creating files in the BASIC or FORTRAN subsystem, FSE assumes the file is a numbered sequenced file. In directives, you can then refer to the sequence numbers rather than to the internal line numbers FSE provides. Refer to the SET NUMBER directive for more information on manipulating sequence numbers.

Using Other Terminals 8

Using Other Terminals 8

This section provides you with worksheets to record the keys on your terminal that correspond to the Viking 721 terminal keys and the default settings of programmable function keys F1 through F16. For example, the NEXT key description appears on the worksheet as:

Viking 721 Key	Your Terminal Key	Function
NEXT	________________	Terminates the input line.

If you were using the worksheet on a CDC 722 terminal, you would fill in the space provided as follows:

Viking 721 Key	Your Terminal Key	Function
NEXT	NEW LINE or CR	Terminates the input line.

Space is also provided for any information not covered in the worksheet that you want to include.

Refer to appendix D for information on corresponding keys for the CDC 722, DEC VT100, Zenith Z19, Heathkit H19, Lear Siegler ADM3A and ADM5, and Tektronix 4115 terminals.

Terminal ____________

Viking 721 Key	Your Terminal Key	Function
FWD	____________	Advances screen one page.
BKW	____________	Moves screen backward one page.
UP	____________	Moves the current line to the top of the screen.
DOWN	____________	Moves the current line to the bottom of the screen.
HELP	____________	Displays the FSE HELP file.
BACK	____________	Returns you to the previously edited file.
⇧ INSRT	____________	Inserts a blank line, allowing you to type in a new line of text.
INSRT	____________	Inserts a blank character into which you type a new character.
⇧ DLETE	____________	Deletes the current line.
DLETE	____________	Deletes the current character.
⇧ CLEAR	____________	Rewrites the entire screen.
HOME	____________	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	____________	Terminates an input line.

Default Programmable Function Key Settings

MARK ______________________________

MRKCHR ______________________________

MOVE ______________________________

COPY ______________________________

INSB ______________________________

DELB ______________________________

FIRST ______________________________

LAST ______________________________

UNDO ______________________________

QUIT ______________________________

LOCNXT ______________________________

LOCATE ______________________________

80COL ______________________________

132COL ______________________________

MIDDLE ______________________________

ENDLIN ______________________________

SPLIT ______________________________

JOIN ______________________________

PARA ______________________________

COPY ______________________________

CENTER ______________________________

NOTES: ____

____

____

Terminal

Viking 721 Key	Your Terminal Key	Function
FWD	______	Advances screen one page.
BKW	______	Moves screen backward one page.
UP	______	Moves the current line to the top of the screen.
DOWN	______	Moves the current line to the bottom of the screen.
HELP	______	Displays the FSE HELP file.
BACK	______	Returns you to the previous edit file.
⇧ INSRT	______	Inserts a blank line, allowing you to type in a new line of text.
INSRT	______	Inserts a blank character into which you type a new character.
⇧ DLETE	______	Deletes the current line.
DLETE	______	Deletes the current character.
⇧ CLEAR	______	Rewrites the entire screen.
HOME	______	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	______	Terminates an input line.

Default Programmable Function Key Settings

MARK ______________________________

MRKCHR ______________________________

MOVE ______________________________

COPY ______________________________

INSB ______________________________

DELB ______________________________

FIRST ______________________________

LAST ______________________________

UNDO ______________________________

QUIT ______________________________

LOCNXT ______________________________

LOCATE ______________________________

80COL ______________________________

132COL ______________________________

MIDDLE ______________________________

ENDLIN ______________________________

SPLIT ______________________________

JOIN ______________________________

PARA ______________________________

COPY ______________________________

CENTER ______________________________

NOTES: ____

____

____

Code Set Conversion A

The code conversion information in this appendix is provided to help you interpret information coded in 6/12-bit display code or 7-bit† ASCII code when it is displayed in 6-bit display code form. The left side of table A-1 lists the 128-character ASCII character set with the corresponding 6-bit display code values. The right side of the table shows the 6/12-bit display code and 7-bit ASCII code characters as they appear when displayed in 6-bit display code format.

Table A-1. Code Conversion Chart

ASCII (128-Character)			6-Bit Display Code		6/12-Bit Display Code		7-Bit ASCII Code Char.
Character	Octal	Hex.	Char.	Octal	Char.	Octal	
NUL	000	00			^5	7640	5:
SOH	001	01			^6	7641	:A
STX	002	02			^7	7642	:B
ETX	003	03			^8	7643	:C
EOT	004	04			^9	7644	:D
ENQ	005	05			^+	7645	:E
ACK	006	06			^-	7646	:F
BEL	007	07			^*	7647	:G
BS	010	08			^/	7650	:H
HT	011	09			^(	7651	:I
LF	012	0A			^)	7652	:J
VT	013	0B			^$	7653	:K
FF	014	0C			^=	7654	:L
CR	015	0D			^sp	7655	:M
SO	016	0E			^,	7656	:N
SI	017	0F			^.	7657	:O
DLE	020	10			^#	7660	:P
DC1	021	11			^[	7661	:Q
DC2	022	12			^]	7662	:R
DC3	023	13			^%	7663	:S
DC4	024	14			^"	7664	:T
NAK	025	15			^_	7665	:U
SYN	026	16			^!	7666	:V
ETB	027	17			^&	7667	:W

Continued on next page

† 7-bit ASCII characters occupy the rightmost 7 bits of a 12-bit field. The leftmost 5 bits are unused.

Table A-1. Code Conversion Chart *(Continued)*

ASCII (128-Character)			6-Bit Display Code		6/12-Bit Display Code		7-Bit ASCII Code
Character	Octal	Hex.	Char.	Octal	Char.	Octal	Char.
CAN	030	18			^'	7670	:X
EM	031	19			^?	7671	:Y
SUB	032	1A			^<	7672	:Z
ESC	033	1B			^>	7673	:0
FS	034	1C			^@	7674	:1
GS	035	1D			^\	7675	:2
RS	036	1E			^ ^	7676	:3
US	037	1F			^;	7677	:4
sp	040	20	sp	55	sp	55	:5
! Exclamation Point	041	21	!	66	!	66	:6
" Quotation Marks	042	22	"	64	"	64	:7
# Number Sign	043	23	#	60	#	60	:8
$ Dollar Sign	044	24	$	53	$	53	:9
% Percent Sign	045	25	%	63	%	63	:+
& Ampersand	046	26	&	67	&	67	:-
' Apostrophe	047	27	'	70	'	70	:*
(Opening Parenthesis	050	28	(	51	(	51	:/
) Closing Parenthesis	051	29	)	52	)	52	:(
* Asterisk	052	2A	*	47	*	47	:)
+ Plus	053	2B	+	45	+	45	:$
, Comma	054	2C	,	56	,	56	:=
- Dash	055	2D	-	46	-	46	:sp
. Period	056	2E	.	57	.	57	:.
/ Slant	057	2F	/	50	/	50	:/
0	060	30	0	33	0	33	:#
1	061	31	1	34	1	34	:[
2	062	32	2	35	2	35	:]
3	063	33	3	36	3	36	:%
4	064	34	4	37	4	37	:"
5	065	35	5	40	5	40	:_
6	066	36	6	41	6	41	:!
7	067	37	7	42	7	42	:&
8	070	38	8	43	8	43	:'
9	071	39	9	44	9	44	:?
: Colon	072	3A	:	00	@D	7404	:<
; Semicolon	073	3B	;	77	;	77	:>
< Less than	074	3C	<	72	<	72	:@
= Equals	075	3D	=	54	=	54	:\
> Greater than	076	3E	>	73	>	73	:^
? Question Mark	077	3F	?	71	?	71	:;

Continued on next page

Table A-1. Code Conversion Chart *(Continued)*

ASCII (128-Character)			6-Bit Display Code		6/12-Bit Display Code		7-Bit ASCII Code Char.
Character	Octal	Hex.	Char.	Octal	Char.	Octal	
@ Commercial At	100	40	@	74	@A	7401	A:
A	101	41	A	01	A	01	AA
B	102	42	B	02	B	02	AB
C	103	43	C	03	C	03	AC
D	104	44	D	04	D	04	AD
E	105	45	E	05	E	05	AE
F	106	46	F	06	F	06	AF
G	107	47	G	07	G	07	AG
H	110	48	H	10	H	10	AH
I	111	49	I	11	I	11	AI
J	112	4A	J	12	J	12	AJ
K	113	4B	K	13	K	13	AK
L	114	4C	L	14	L	14	AL
M	115	4D	M	15	M	15	AM
N	116	4E	N	16	N	16	AN
O	117	4F	O	17	O	17	AO
P	120	50	P	20	P	20	AP
Q	121	51	Q	21	Q	21	AQ
R	122	52	R	22	R	22	AR
S	123	53	S	23	S	23	AS
T	124	54	T	24	T	24	AT
U	125	55	U	25	U	25	AU
V	126	56	V	26	V	26	AV
W	127	57	W	27	W	27	AW
X	130	58	X	30	X	30	AX
Y	131	59	Y	31	Y	31	AY
Z	132	5A	Z	32	Z	32	AZ
[Opening Bracket	133	5B	[	61	[	61	A0
\ Reverse Slant	134	5C	\	75	\	75	A1
] Closing Bracket	135	5D	]	62	]	62	A2
^ Circumflex	136	5E	^	76	@B	7402	A3
_ Underline	137	5F	_	65	_	65	A4
` Grave Accent	140	60	@	74	@G	7407	A5
a	141	61			^A	7601	A6
b	142	62			^B	7602	A7
c	143	63			^C	7603	A8
d	144	64			^D	7604	A9
e	145	65			^E	7605	A+
f	146	66			^F	7606	A-
g	147	67			^G	7607	A*

Continued on next page

Table A-1. Code Conversion Chart *(Continued)*

ASCII (128-Character)			6-Bit Display Code		6/12-Bit Display Code		7-Bit ASCII Code
Character	Octal	Hex.	Char.	Octal	Char.	Octal	Char.
h	150	68			^H	7610	A/
i	151	69			^I	7611	A(
j	152	6A			^J	7612	A)
k	153	6B			^K	7613	A$
l	154	6C			^L	7614	A=
m	155	6D			^M	7615	Asp
n	156	6E			^N	7616	A,
o	157	6F			^O	7617	A.
p	160	70			^P	7620	A#
q	161	71			^Q	7621	A[
r	162	72			^R	7622	A]
s	163	73			^S	7623	A%
t	164	74			^T	7624	A"
u	165	75			^U	7625	A_
v	166	76			^V	7626	A!
w	167	77			^W	7627	A&
x	170	78			^X	7630	A'
y	171	79			^Y	7631	A?
z	172	7A			^Z	7632	A<
{ Opening Brace	173	7B	[	61	^0	7633	A>
| Vertical Line	174	7C	\	75	^1	7634	A@
} Closing Brace	175	7D	]	62	^2	7635	A\
~ Tilde	176	7E	^	76	^3	7636	A^
DEL	177	7F			^4	7637	A;

NOTE: sp represents a space.

Error Recovery and Diagnostic Messages B

Appendix B contains procedures for recovering from errors that either cause you to lose the connection to NOS or, when screen editing, cause your screen to become garbled. This appendix also contains alphabetical listings of FSE diagnostic messages.

Error Recovery

This subsection describes how to recover your FSE session if one of the following errors occurs.

- The screen is garbled due to communication errors.
- The connection to NOS is lost.

Recovering from Communication Errors

If communication errors cause your screen to become garbled, you can rewrite the current screen as follows:

- If you are on a Viking 721 terminal, press:

  ⇧ CLEAR + NEXT

- If you are on a CDC 722 terminal, press:

  EOP + NEW LINE

- On any terminal, press:

  HOME

  (or the equivalent key) to position the cursor on the FSE directive line. Then enter:

  ```
  S S
  ```

Recovering When You Lose the NOS Connection

The following procedure restores your FSE session should you lose the connection to NOS.

1. Log in to NOS.

```
WELCOME TO THE NOS SOFTWARE SYSTEM.
COPYRIGHT CONTROL DATA 1978, 198x.

yy/mm/dd   13.20.42  L3FT1
CDC NETWORK OPERATING SYSTEM NOS 2
FAMILY:    familyname,username,password
JSN: ABVU, NAMIAF
```

2. When the following appears, enter the job sequence name (JSN) of the job to recover. In this example, only job ABVJ can be recovered.

```
     RECOVERABLE JOB(S)

     JSN    UJN       STATUS       TIMEOUT

     ABVJ   AKVA      SUSPENDED    29 MIN.

  ENTER  GO  TO CONTINUE CURRENT JOB,
         RELIST  TO LIST RECOVERABLE JOBS,
         OR DESIRED JSN:  abvj
```

3. Do not enter GO when the following message appears.

```
JSN: ABVJ  SYSTEM: BATCH    SRU:      2.679
STATUS:  FSE,MYFILE,A,G.
CHARACTER SET: ASCII    MODES: PROMPT ON
INPUT REQUESTED.  ENTER GO TO CONTINUE.
```

   Instead, press:

```
NEXT
```

   The following FSE prompt appears.

```
??
```

4. Respond to the prompt with:

```
S S
```

   This returns you to where you left off in your FSE session.

Refer to the *NOS Reference Set, Volume 2, Guide to System Usage*, for detailed information on NOS login and job recovery procedures.

Diagnostic Messages

FSE diagnostic messages are grouped in two categories: user messages and internal messages. User messages appear when you make a mistake while using FSE. Internal messages appear because FSE made a mistake; they are always preceded by a line that says `FSE INTERNAL ERROR` and followed by `CONTACT SOFTWARE SUPPORT.`

The user messages are listed here followed by the internal messages. If you receive a message not listed, either refer to the *NOS Reference Set, Volume 3, System Commands*, or contact Central Software Support.

User Messages

`BATCH JOBS MUST BE ERROR FREE.`

A syntax error occurred on a noninteractive FSE session. Correct the job and try again.

`CANNOT CHANGE READ-ONLY FILE.`

You tried to change a file that you are not allowed to change.

`CHARACTER SEQUENCE NOT RECOGNIZED.`

Your input is not recognized by FSE. Check the spelling and try again.

`DIRECTIVE CANCELLED BY FUNCTION KEY.`

You pressed a function key before a directive had finished executing. The function key cancels the directive and executes the function. No action on your part is required.

`EMPTY FILE.`

You have specified a file that contains no lines. Try again with the correct file name.

`END OF LINE BEYOND EDGE OF SCREEN.`

You tried to position the cursor to the end of a line that extends beyond the edge of the screen. Use the appropriate key or directive to view the wide line.

ERROR IN THIS PROCEDURE LINE.

The procedure line displayed contains an error. Correct the error and try again.

FILE MUST BE ON DISK.

The file you specified is not on a mass storage device and could not, therefore, be located. Try again with a valid file name.

FILE MUST NOT CONTAIN SEQUENCE NUMBERS.

The directive you specified can be performed only on files without sequence numbers. Try another directive or file.

FILE NAME IN USE.

The file name you specified is already in use in the work file. Select a different file name and try again.

INTERNAL ERROR.....

Refer to the following *Internal Messages* subsection.

INVALID FILE NAME.

You specified an invalid file name. Check the spelling of the file name and try again.

KEYWORD MUST FOLLOW SET.

You entered an invalid keyword parameter following the SET directive. Check the syntax of the SET directive and try again.

LINE INCREMENT VALUE TOO LARGE.

The value you specified on the SET INCREMENT directive is too large for the specified file. Try again with a smaller increment.

MARKS CANCELLED.

You cancelled the previously set marks by either pressing UNDO or entering the UNDO directive.

MISSING REPLACEMENT STRING.

You entered a REPLACE directive that did not contain a replacement string. Try again with a specified string.

```
MUST SPECIFY FILE NAME.
```

The FSE directive you entered did not contain a valid file name. Try again with a valid file name.

```
MUST SPECIFY YES OR NO.
```

You specified a keyword parameter other than YES or NO. Try again with YES or NO specified.

```
NOT ENOUGH ROOM FOR INSERTION.
```

There is not enough room for you to insert any lines. Resequence the line numbers and try again.

```
NOT FOUND.
```

The string you specified on a LOCATE or REPLACE directive was not found in the file. Check the spelling of the string and try again.

```
ONLY ONE RANGE ALLOWED.
```

You specified more than one range. Check the syntax of the desired directive and try again.

```
OUT OF BOUNDS.
```

The range you specified is out of bounds. Specify a smaller range and try again.

```
PARAMETER NOT VALID FOR THIS DIRECTIVE.
```

You have specified a keyword parameter that FSE does not recognize. Check the command syntax and try again.

```
PFN BUSY OR NOT FOUND, USING LOCAL FILE.
```

FSE could not access the permanent file specified on the FSE command because it was busy or the file does not exist. FSE then uses the local copy of the file, if any. Check the file name and try again.

```
PLEASE RE-ENTER INPUT.
```

You must wait until one directive or function has completed before you can enter another. Try again.

```
PROCEDURE NOT FOUND.
```

FSE could not find the procedure specified. Check the file on which the procedure resides, check the spelling, and try again.

`QUIT IS REQUIRED FOR BATCH JOBS.`

You did not include a QUIT directive in a noninteractive FSE session. Add a QUIT directive and try again.

`RANGE MUST BE IN SAME FILE.`

The range you specified spanned more than one file. Try again with the range limited to one file.

`RESERVED FILE.`

You tried to edit a file that is protected from editing. Start FSE again with a valid file name.

`RETURN MUST FOLLOW FUNCTION KEY.`

You must enter a carriage return after pressing the specified function key. Try again, pressing RETURN after the function key.

`SCREEN MODE REQUIRED.`

You tried an editing function that works only in screen editing mode. Either try another function or set screen mode and try again.

`SET UNDO YES - TO ENABLE UNDO.`

You tried to enter the UNDO directive while it is not enabled. Enable UNDO (using the SET UNDO YES directive) and try again.

`STRING GREATER THAN 80 CHARACTERS.`

You entered a string containing a line longer than the allowable limit (80 characters).

`STRING NOT ALLOWED.`

You entered a string when none is required. Try again without the string.

`SYSTEM INTERRUPT, PROCEED NOW.`

A forced transaction from multiuser FSE to single-user FSE has occurred. This is not a result of anything you did. Resume editing. The effects of the previous directive may be cancelled.

`TAB FIELD ORDINAL OUT OF BOUNDS.`

You specified a tab field that was out of range. Try again with a valid tab field.

TAB STOP OUT OF BOUNDS.

You specified a tab field that is out of range. Try again with a valid tab field entry.

TERMINAL MODEL NAME UNKNOWN.

You entered an incorrect terminal model name. Check the terminal model and try again.

TOO MANY PARAMETERS.

You specified too many parameters on one directive. Take off a few parameters and try again.

UNKNOWN CONTROL KEY.

FSE detected a bad code sequence you entered. Try again.

UNKNOWN DIRECTIVE.

You entered a directive that FSE did not recognize. Check the spelling and syntax and try again.

UNKNOWN FSE COMMAND OPTION.

You specified an option on the FSE command that FSE did not recognize. Check the FSE command format and try again.

UNSUPPORTED FUNCTION KEY.

FSE does not support the key you pressed. Check the key you want and try again.

VALUE MUST BE NUMERIC.

You entered a nonnumeric value for a keyword parameter that must be numeric. Enter a numeric value and try again.

WIDE LINE.

You specified a range of lines containing a line longer than the limit. To complete the processing of the line range, increase the line width limit.

Internal Messages

The following messages are always preceded by the line
`FSE INTERNAL ERROR.`

NOTE

When an FSE internal error message occurs, you should attempt to repeat the procedure that produced the message. If the message reappears, contact Central Software Support.

```
ALL WORKFILE BUFFERS ARE FULL.
```

FSE encountered a full working buffer while trying to store data.

```
ALTERED PRU WAS NOT REWRITTEN.
```

FSE failed to rewrite an altered PRU before reusing the buffer. Contact Central Software Support.

```
BAK BEFORE START OF FILE.
```

FSE attempted to access a line located before the first line of the file. Contact Central Software Support.

```
DIRECTORY BLOCK POINTER TOO LARGE.
```

The directory block pointer is greater than the value allowed. Contact Central Software Support.

```
DIRECTORY BLOCK POINTER TOO SMALL.
```

The directory block pointer is less than the value allowed. Contact Central Software Support.

```
FILE MUST BE ON DISK.
```

The file specified on the FSE command is not on a mass storage device. Try again with a valid file name.

```
FILE POSITION STACK OVERFLOWED.
```

The stack that monitors the file position overflowed. Contact Central Software Support.

```
FILE POSITION STACK UNDERFLOWED.
```

The stack that monitors the file position underflowed. Contact Central Software Support.

FILE SIZE INCORRECT.

The file size (EOI sector) reported by CIO conflicts with FSE. Contact Central Software Support.

FILE TOO LARGE.

The file you are attempting to edit is too large for FSE to handle. Contact Central Software Support.

FWD BEYOND END OF FILE.

FSE attempted to access a line located after the last line of the file. Contact Central Software Support.

LINE NOT FOUND IN FILE.

FSE attempted to access a line that does not exist in the file. Contact Central Software Support.

NO FILE NAME ON FSE COMMAND.

You entered an FSE command without specifying a file name. Reenter the FSE command with a valid file name.

OLD SECTOR MUST BE WRITTEN.

FSE attempted to allocate a new sector without first flushing the old sector to disk. Contact Central Software Support.

OUT-OF-BOUNDS PRU ADDRESS ON READ.

FSE attempted to read a sector address that is out of bounds. Contact Central Software Support.

PREVIOUS VERSION OF WORKFILE.

FSE and the workfile versions do not match. Contact Central Software Support.

PRU CONTENT INCORRECT.

The content of the PRU read does not match the expected value. Contact Central Software Support.

RANDOM ADDRESS INCORRECT.

The random address returned by CIO after a random rewrite conflicts with FSE. Contact Central Software Support.

RANDOM READ LIST INCORRECT.

The read list did not contain the sector FSE was trying to access. Contact Central Software Support.

REENTRANT DATA STACK OVERFLOWED.

The general purpose data stack for re-entrant code overflowed. Contact Central Software Support.

REENTRANT DATA STACK UNDERFLOWED.

The general purpose data stack for re-entrant code underflowed.

RPHRLS LIST BUFFER IS FULL.

The list buffer for the RPHRLS macro is full. Contact Central Software Support.

SECTOR CANNOT BE DEALLOCATED.

FSE attempted to deallocate a block that was still valid. Contact Central Software Support.

STATUS FLAGS SHOULD HAVE BEEN CLEARED.

The content of the PRU read includes status flags that should have been cleared previously. Contact Central Software Support.

TRANSFER TO THE BUFFER IS INCOMPLETE.

The buffer transfer on a COPY or MOVE is incomplete. Contact Central Software Support.

UNABLE TO WRITE FROM BUFFER.

FSE was unable to write and empty the circular buffer. Contact Central Software Support.

UNKNOWN TERMINAL MODEL NUMBER.

The terminal model number specified is not defined.

WORKFILE BUFFER ALLOCATED TWICE.

FSE encountered a sector written in two different areas of the work file. Contact Central Software Support.

WORKFILE BUFFER CONTAINS WRONG TEXT.

FSE encountered a working buffer with incorrect text. Inform Central Software Support.

WORKFILE BUFFER EMPTY ON READ.

FSE encountered an empty circular buffer while reading the workfile. Contact Central Software Support.

WORKFILE BUFFER FULL ON READ.

FSE encountered a full circular buffer while reading the workfile. Contact Central Software Support.

WORKFILE BUFFER FULL ON WRITE.

FSE encountered a full circular buffer while writing the workfile. Contact Central Software Support.

WORKFILE IS NOT IN A RESUMABLE STATE.

FSE could not recognize the contents of the workfile. Contact Central Software Support.

Glossary C

A

ASCII

American National Standard Code for Information Interchange. A 7-bit code representing a prescribed set of 128 characters including both uppercase and lowercase letters.

ASCII Mode

Use of the American National Standard Code for Information Interchange 128-character set. It includes both uppercase and lowercase letters.

Attach

The process of retrieving a direct access permanent file to use at your terminal by specifying the proper permanent file identification and, if required, passwords.

B

BACK

The FSE directive that returns you to a previously edited file, or allows you to exit the HELP file.

BASIC

1. Beginner's All-Purpose Symbolic Instruction Code, an elementary programming language.
2. The subsystem that uses the BASIC compiler.

Batch Job

Instructions and data submitted as a complete unit without further intervention on your part.

Batch Mode

Mode in which instructions and data are submitted as complete units without further intervention on your part.

Break

An interruption in the data stream. A user break (a break normally entered by you at your interactive terminal) stops delivery of a message or output from the host.

C

CDC Standard Keys

Keys that perform functions on nearly all applications and can be performed on all supported terminals.

Character

Any alphabetic, numeric, or special symbol that can be encoded. This term applies to the graphic characters for an input or output device, and to uniquely encoded control characters used by a terminal.

Character Set

A prescribed set of characters, specified at the time the operating system is installed. FSE supports the 128-character ASCII character set and also subsets such as the CDC 64-bit character set.

Command

An instruction you give to NOS that causes it to perform an operation.

Current Character

The character on which the cursor is positioned.

Current Editing Session

The editing session started with the latest entry of the FSE command that included parameters.

Current Line

The line on which the cursor is positioned.

Current Screen

The text appearing on your screen.

Current Terminal Session

The terminal session started with the latest physical connection to the system.

D

Default

A system-supplied value used when you do not supply a value.

Direct Access File

A NOS permanent mass storage file that can be attached to your terminal session. All changes to this file are made on the file itself, rather than on a temporary copy of the file (compare with *Indirect Access File*).

Directive

An instruction to FSE.

Display Code

A 6-bit character code set used to represent alphanumeric and special characters. Display code includes only uppercase letters.

E

Editing Keys

Keys, such as INSRT , DLETE ,ERASE, and others, whose functions are predetermined and usually performed by software at your terminal. Contrast with *Programmable Function Keys*.

Editing Session

The time elapsing from when you start FSE (by entering the FSE command with parameters) to when you stop FSE (by either logging off NOS or entering another FSE command with parameters). You can restart your previous editing session by entering the FSE command without parameters.

Ellipsis String

A parameter delimited by strings enabling you to specify longer strings. The first string delimiter must be on the same line as the last string delimiter.

EOF (End-of-File)

A boundary within a sequential file. EOF is not necessarily the end of a file that can be referenced by name. The actual end of a named file is defined by EOI. In the product set manuals (FORTRAN and so forth), an end-of-file is also referred to as an end-of-partition.

EOI (End-of-Information)

The end of data on a file. Information appearing after this point is not considered file data. CDC CYBER Record Manager defines end-of-information in terms of file residency and organization.

EOR (End-of-Record)

An indicator that marks the end of a logical record. In product set manuals (FORTRAN, and so on), end-of-record is also referred to as end-of-section.

F

File

A collection of information referred to by a file name (seven or fewer alphanumeric characters). You can create a file at your terminal or retrieve a file from permanent file storage for use during a terminal session.

File Header

In FSE screen mode, the file header is the line containing the following information.

- File name and type.
- Range of lines currently displayed.
- Total number of lines in the file.

File Name

Name assigned to a file; it can contain from one through seven letters and digits, the first of which must be alphabetic.

FORTRAN

1. Formula Translation, a high-level language consisting of symbols and statements that can be used to create a program closely resembling mathematical notation.
2. The subsystem that uses the FORTRAN Version 5 compiler.

FSE

The NOS command that starts the Full Screen Editor.

FSE Procedures

Groups of directives that perform the FSE functions and can be grouped under one name.

FSEPROC

The library file in which most FSE procedures reside.

Full Screen Editor

An editing utility that enables you to edit files either screen-by-screen or line-by-line.

H

HELP File

An online reference aid that lists the FSE directives and their parameters.

I

Indirect Access File

A NOS permanent file that you access by making a temporary copy of it (GET or OLD command). You create or alter it by saving or substituting the contents of an existing temporary file (QUIT REPLACE directive).

Input

Information flowing from your terminal to the host mainframe.

K

Keyword

A type of parameter used in an FSE directive, usually followed by a value.

L

Line Mode

The FSE editing mode in which a line of the file is the basic unit of operation.

Local File

Any file assigned to your terminal session; this includes all temporary (indirect access permanent) files, all direct access permanent files, and all files that are not permanent.

Login

The procedure used at an interactive terminal to gain access to the system.

Logoff

The procedure used to end a terminal session.

M

Micros

Commands within procedures or directive strings that allow you to specify the current character, word, line, or file as a parameter.

Modification Character

A special character (#, &, !, or ^ characters) used with the ALTER directive to change text.

N

NOS

Network Operating System.

O

Output

Information flowing from the host mainframe to your terminal.

P

Pad Shifted

Terminal characteristic that enables shifted keypad functions.

Paragraph

A group of lines not containing a blank line.

Parameter

A value following a command name or directive name that alters the behavior of the command or directive.

Permanent File

A file that does not disappear when you log off the system. There are two types of permanent files on NOS: indirect access and direct access.

You access indirect access permanent files indirectly (hence, the name). That is, NOS makes a copy of your permanent file, which you can then use without affecting the permanent copy.

Direct access files are accessed directly and any changes you make are made to the actual permanent file.

Usually, indirect access files are used for smaller files and direct access files are used for very large files.

Procedure

A user-defined set of instructions that are referenced by name. Within FSE you can use procedures that consist of FSE directives.

Programmable Function Keys

Keys defined either by FSE or by you to perform FSE functions.

Q

QUIT

The FSE directive used to exit the editor, exit the TEACH file, or end a procedure.

R

Record

Divisions within a file. Within FSE, the FSEPROC file contains procedures. Each of these procedures is a record terminated by an (EOR) marker or a QP directive.

Restart

Re-entering FSE at the point you exited. To restart FSE, enter the FSE command without parameters.

S

Screen Mode

The FSE editing mode in which a file is edited with a page of text as the basic unit of operation.

Soft Tab Character

A software character instructing FSE to move text to the next tab setting.

String

Any combination of alphanumeric characters bounded by delimiters.

T

TEACH File

A tutorial designed to help the user become familiar with the Viking 721 terminal and the basics of screen editing.

Temporary File

A file that is assigned to your terminal session and vanishes when you either release it or end your terminal session. Local copies of indirect access permanent files are temporary files; direct access permanent files assigned to your terminal session are not temporary files.

Terminal Definition Utility (TDU)

The utility that allows you to define your terminal's characteristics to NOS.

Terminal Session

The lapse between the time you physically connect the terminal to the system and the time you log out.

Typematic

A terminal characteristic that causes keys to automatically repeat their functions when held down.

U

UNDO

An editing directive that allows you to delete changes you have made in the reverse order that they were made. UNDO should be turned off for batch processing (SET UNDO NO).

W

Word

A string of text delimited by nonalphanumeric characters or blank characters.

Terminal Support Information D

This appendix provides you with the information you will need to run FSE in screen mode on the following terminals.

- CDC 722.
- DEC VT100.†
- Zenith Z19 or Heathkit H19.†
- Lear Siegler ADM3A and ADM5.†
- Tektronix 4115.†

Information on these terminals includes:

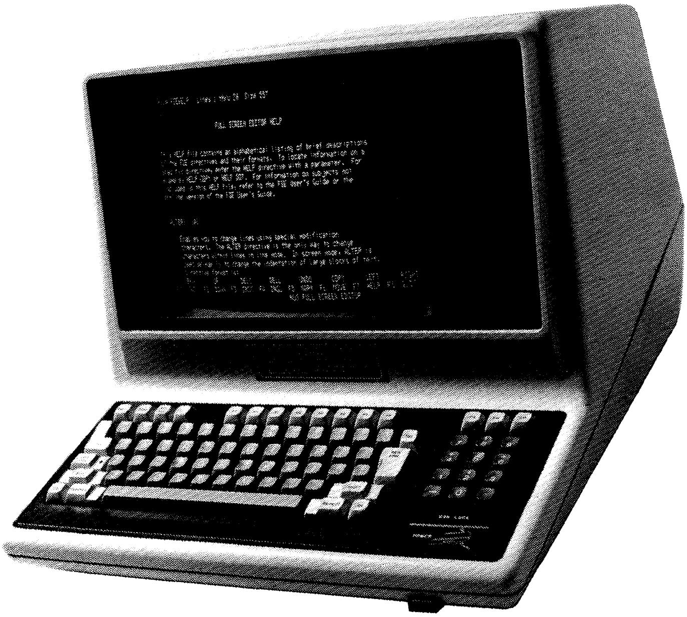
- The keys equivalent to the Viking 721 editing and CDC standard keys.
- The default settings for the programmable function keys.
- Other miscellaneous information.

† The network control character (CT) for this terminal should be something other than ESC. The terminal uses ESC sequences for function key definitions. To change the network control character, enter:

```
TRMDEF,CT=value
```

For details, refer to the *NOS Reference Set, Volume 3, System Commands.*

CDC 722



Equivalent Keys

Viking 721 Key	Equivalent on CDC 722	Function
FWD	F1 + NEW LINE	Advances screen one page.
BKW	SHIFT F1 + NEW LINE	Moves screen backward one page.
UP	F2 + NEW LINE	Moves current line to top of screen.
DOWN	SHIFT F2 + NEW LINE	Moves current line to bottom of screen.
HELP	F7 + NEW LINE	Displays the FSE HELP file.
BACK	BACK directive	Returns you to the previously edited file.
⇧ INSRT	F4 + NEW LINE	Inserts a blank line allowing you to type in a new line of text.
INSRT	F3 + NEW LINE	Inserts a blank character over which you type the new character.
⇧ DLETE	SHIFT F4 + NEW LINE	Deletes the current line.
DLETE	SHIFT F3 + NEW LINE	Deletes the current character.
CLEAR	EOL	Deletes all characters from the cursor to the end of the line.
⇧ CLEAR	SHIFT EOL	Rewrites the entire screen.
HOME	SHIFT HOME	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	NEW LINE or CR	Terminates an input line.
↑ 8 ← 4 → 6 ↓ 2	SHIFT ↑ 8 ← 4 → 6 ↓ 2	Move cursor around on screen.

Default Programmable Function Keys for the CDC 722

Key	Description
F1 BKW FWD	F1 FWD moves forward one page in the file. Shifted, F1 BKW moves backward one page in the file.
F2 LINEDN LINEUP	F2 LINEUP moves the current line to the top of the screen. Shifted, F2 LINEDN positions the current line to the bottom of the screen.
F3 DELC INSC	F3 INSC inserts a blank at the current character (you can type a new character over the blank). Shifted, the F3 DELC key deletes the current character. You can press the F3 key several times before pressing NEW LINE to perform delete or insert more than one character. It is not until you press NEW LINE that the results are shown.
F4 DELL INSL	F4 INSL inserts a blank line over which you can type new text. Shifted, F4 DELL deletes the current line. You can press the F4 key several times before pressing NEW LINE to insert or delete more than one line. It is not until you press NEW LINE that the results are shown.
F5 UNDO MARK	F5 MARK marks a line or lines for later use with another directive. These marked lines are not in inverse video display as on the Viking 721. Shifted, F5 UNDO undoes the previous change to your file.

Key	Description
F6 COPY MOVE	F6 MOVE moves any marked lines or characters to the current line or character. Shifted, F6 COPY copies any marked lines or characters to the current line or character.
F7 LEFT HELP	F7 HELP displays the FSE HELP file. Shifted, F7 LEFT moves your view of the file to the left.
F8 RIGHT QUIT	F8 QUIT stops FSE without making any changes permanent. Shifted, F8 RIGHT moves your view of the file to the right.
F9 ENDLIN	F9 ENDLIN moves the cursor to the end of the current line.

NOTE

You must press NEW LINE or CR after pressing a programmable function key.

Miscellaneous Information

Do not use the TAB key to insert tabs. Instead, use the soft tab character (the default character is \ ).

DEC VT100

Equivalent Keys

Viking 721 Key	Equivalent on DEC VT100†	Function
FWD	1 + RETURN	Advances screen one page.
BKW	PF1 + RETURN	Moves screen backward one page.
DOWN	PF2 + RETURN	Moves current line to bottom of screen.
UP	2 + RETURN	Moves current line to top of screen.
HELP	7 + RETURN	Displays the FSE HELP file.
BACK	BACK directive	Returns you to the previously edited file.
⇧ INSRT	4 + RETURN	Inserts a blank line allowing you to type in a new line of text.
INSRT	3 + RETURN	Inserts a blank character over which you type the new character.
⇧ DLETE	PF4 + RETURN	Deletes the current line.
DLETE	PF3 + RETURN	Deletes the current character.
⇧ CLEAR	. + RETURN	Rewrites the entire screen.
HOME	ENTER + RETURN	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	RETURN	Terminates an input line.

† All keys listed here (except RETURN) are located on the keypad to the right of the main keyboard.

Default Programmable Function Keys for the DEC VT100

Key	Description
BKW F1 FWD	F1 FWD moves forward one page in the file. Shifted, F1 BKW moves backward one page in the file.
LINEDN F2 LINEUP	F2 LINEUP moves the current line to the top of the screen. Shifted, F2 LINEDN positions the current line to the bottom of the screen.
DELC F3 INSC	F3 INSC inserts a blank at the current character (you can type a new character over the blank). Shifted, F3 DELC deletes the current character. You can press the F3 key several times before pressing RETURN to perform delete or insert more than one character. It is not until you press RETURN that the results are shown.
DELL F4 INSL	F4 INSL inserts a blank line over which you can type new text. Shifted, F4 DELL deletes the current line. You can press the F4 key several times before pressing RETURN to insert or delete more than one line. It is not until you press RETURN that the results are shown.
UNDO F5 MARK	F5 MARK marks a line or lines for later use with another directive. These marked lines are in inverse video display as on the Viking 721. Shifted, F5 UNDO undoes the previous change to your file.

Key	Description
COPY F6 MOVE	F6 MOVE moves any marked lines or characters to the current line or character. Shifted, F6 COPY copies any marked lines or characters to the current line or character.
HOME F7 HELP	F7 HELP displays the FSE HELP file. Shifted, F7 HOME positions the cursor on the FSE directive line.
CLEAR F8 QUIT	F8 QUIT stops FSE without making any changes permanent. Shifted, F8 CLEAR clears your screen.
F9 ENDLIN	F9 ENDLIN moves the cursor to the end of the current line.

NOTES

- You must press RETURN after pressing a programmable function key.
- The unshifted function keys F1 through F9 are the keypad keys 1 through 9. (Within FSE, the keypad cannot be used for numeric values).
- The shifted function keys are the following keypad keys.

Function Key	Keypad Key
Shifted F1	PF1
Shifted F2	PF2
Shifted F3	PF3
Shifted F4	PF4
Shifted F5	−
Shifted F6	,
Shifted F7	ENTER
Shifted F8	.
Shifted F9	0

Zenith Z19 or Heathkit H19

Equivalent Keys

Viking 721 Key	Equivalent on Zenith Z19/ Heathkit H19	Function
(FWD)	(f1) + (RETURN)	Advances screen one page.
(BKW)	(f2) + (RETURN)	Moves screen backward one page.
(UP)	(f3) + (RETURN)	Moves current line to top of screen.
(DOWN)	(f4) + (RETURN)	Moves current line to bottom of screen.
(HELP)	■ + (RETURN)	Displays the FSE HELP file.

Viking 721 Key	Equivalent on Zenith Z19/ Heathkit H19	Function
BACK	BACK directive	Returns you to the previously edited file.
⇧ INSRT	IL †	Inserts a blank line allowing you to type in a new line of text. When you insert lines, the function key prompts move according to the number of lines inserted. To align the prompts properly, press RETURN . This key does not require you to press RETURN afterward.
INSRT	IC †	When you press IC, it puts the terminal into insert mode. While insert mode is on, pressing any character moves the existing text to the right and inserts the new character. Insert mode is cancelled when you either press IC a second time or press RETURN.
⇧ DLETE	DL †	Deletes the current line. When you delete lines, the function key prompts move according to the number of lines deleted. To align the prompts properly, press RETURN. This key does not require you to press RETURN afterward.
DLETE	DC †	Deletes the current character.
⇧ CLEAR	SHIFT ERASE + RETURN	Rewrites the entire screen.
HOME	HOME †	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	RETURN	Terminates an input line.

† Located on the numeric keypad to the right of the main keyboard.

Default Programmable Function Keys for the Zenith Z19/Heathkit H19

Key	Description
F1 MARK FWD	F1 FWD moves forward one page in the file. Shifted, F1 MARK marks a line or lines to be used with another function or directive.
F2 MRKCHR BKW	F2 BKW moves backward one page in the file. Shifted, F2 MRKCHR marks a character or characters for use with another function or directive.
F3 LINEUP	F3 LINEUP moves the current line to the top of the screen.
F4 COPY LINEDN	F4 LINEDN positions the current line to the bottom of the screen. Shifted, F4 COPY copies any marked lines or characters to the current line or character.
F5 MOVE ENDLIN	F5 ENDLIN moves the cursor to the end of the current line. Shifted, F5 MOVE moves any marked lines or characters to the current line or character.
F6 UNDO	F6 UNDO undoes the previous change to your file.
F7 LEFT QUIT	F7 QUIT exits FSE without making changes to your file permanent. Shifted, F7 LEFT moves your view of the file to the left.
F8 RIGHT HELP	F8 HELP accesses the FSE HELP file. Shifted, F8 RIGHT moves your view of the file to the right.

NOTES

- You must enter RETURN after pressing a programmable function key.
- Unshifted function keys F1 through F5 are keyboard keys f1 through f5. Unshifted function keys F6 through F8 are the following keyboard keys.

Function Key	Keyboard Key
F6	■ (blue square)
F7	■ (red square)
F8	□ (white square)

- Shifted function keys F1 through F8 are the shifted numbers on the keypad to the right of the main keyboard. For example, to enter a shifted F3, press:

  SHIFT 3 + RETURN

  When not shifted, keypad keys 1 through 9 allow you to move the cursor, insert, and delete. (The keypad cannot be used for numerals.)

- The shifted keypad keys 0, ., and ENTER are the shifted function keys F10, F11, and F12.

Miscellaneous Information

The Z19 hardware has tabs set every eighth column beginning with 1. These tabs are set at columns:

1 9 17 25 33 41 49 57 65 73

To specify tabs other than these, use the tab character (default is \ ).

Lear Siegler ADM3A

Equivalent Keys

Viking 721 Key	Equivalent on Lear Siegler ADM3A	Function
FWD	ESC + 1 + RETURN	Advances screen one page.
BKW	ESC + SHIFT 1 + RETURN	Moves screen backward one page.
UP	ESC + 2 + RETURN	Moves current line to top of screen.

Viking 721 Key	Equivalent on Lear Siegler ADM3A	Function
DOWN	ESC + SHIFT 2 + RETURN	Moves current line to bottom of the screen.
HELP	ESC + 7 + RETURN	Displays FSE HELP file.
BACK	BACK directive	Returns you to previously edited file.
⇧ INSRT	ESC + 4 + RETURN	Inserts a blank line allowing you to type in a new line of text.
INSRT	ESC + 3 + RETURN	Inserts a blank character over which you can type the new character.
⇧ DLETE	ESC + SHIFT 4 + RETURN	Deletes the current line.
DLETE	ESC + SHIFT 3 + RETURN	Deletes the current character.
⇧ CLEAR	ESC + 0 + RETURN	Rewrites the entire screen.
HOME	CTRL HOME	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	RETURN	Terminates an input line.
← 4	CTRL ← H	Moves cursor left.
↓ 2	CTRL ↓ J	Moves cursor down.
↑ 8	CTRL ↑ K	Moves cursor up.
→ 6	CTRL → L	Moves cursor right.

Default Programmable Function Keys for the Lear Siegler ADM3A

Key	Description
BKW F1 FWD	F1 FWD moves forward one page in the file. Shifted, F1 BKW moves backward one page in the file.
LINEDN F2 LINEUP	F2 LINEUP moves the current line to the top of the screen. Shifted, F2 LINEDN moves the current line to the bottom of the screen.
DELC F3 INSC	F3 INSC inserts a blank at the current character (you can type a new character over the blank). Shifted, F3 DELC deletes the current character. You can press F3 several times before pressing (RETURN) to delete or insert more than one character. It is not until you press (RETURN) that the results are shown.
DELL F4 INSL	F4 INSL inserts a blank line over which you can type new text. Shifted, F4 DELL deletes the current line. You can press the F4 key several times to insert or delete more than one line. It is not until you press (RETURN) that the results are shown.
UNDO F5 MARK	F5 MARK marks a line or lines for later use with another function. Shifted, F5 UNDO undoes previous changes to your file.
COPY F6 MOVE	F6 MOVE moves any marked lines or characters to the current line or character. Shifted, F6 COPY copies any marked lines or characters to the current line or character.
LEFT F7 HELP	F7 HELP accesses the FSE HELP file. Shifted, F7 LEFT moves your view of the file to the left.
RIGHT F8 QUIT	F8 QUIT exits FSE without making changes to your file permanent. Shifted, F8 RIGHT moves your view of the file to the right.

NOTES

- You must press RETURN after pressing a programmable function key.
- Function keys F1 through F15 are the top row of keyboard keys 1 through ^. Each must be preceded by ESC . For example, to enter an F3, press:

  ESC + 3 + RETURN

  To use the shifted functions of these keys, include the SHIFT key.

  ESC + SHIFT 3 + RETURN
- 0 and SHIFT 0 perform the same function, so in effect there is no shifted F10 function.

Miscellaneous Information

To specify tabs, use the soft tab character (the default character is \).

Lear Siegler ADM5

Equivalent Keys

Viking 721 Key	Equivalent on Lear Siegler ADM5	Function
FWD	ESC + 1 + RETURN	Advances screen one page.
BKW	ESC + SHIFT 1 + RETURN	Moves screen backward one page.
UP	ESC + 2 + RETURN	Moves current line to top of screen.
DOWN	ESC + SHIFT 2 + RETURN	Moves current line to bottom of the screen.
HELP	ESC + 7 + RETURN	Displays FSE HELP file.
BACK	BACK directive	Returns you to previously edited file.
⇧ INSRT	ESC + 4 + RETURN	Inserts a blank line allowing you to type in a new line of text.
INSRT	ESC + 3 + RETURN	Inserts a blank character over which you can type the new character.
⇧ DLETE	ESC + SHIFT 4 + RETURN	Deletes the current line.
DLETE	ESC + SHIFT 3 + RETURN	Deletes the current character.
⇧ CLEAR	PAGE + RETURN	Rewrites the entire screen.
CLEAR	LINE	Clears a line from the current character to the end of the line.
HOME	HOME	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	RETURN	Terminates an input line.

Default Programmable Function Keys for the Lear Siegler ADM5

Key	Description
BKW F1 FWD	F1 FWD moves forward one page in the file. Shifted, F1 BKW moves backward one page in the file.
LINEDN F2 LINEUP	F2 LINEUP moves the current line to the top of the screen. Shifted, F2 LINEDN moves the current line to the bottom of the screen.
DELC F3 INSC	F3 INSC inserts a blank at the current character (you can type a new character over the blank). Shifted, F3 DELC deletes the current character. You can press F3 several times before pressing RETURN to delete or insert more than one character. It is not until you press RETURN that the results are shown.
DELL F4 INSL	F4 INSL inserts a blank line over which you can type new text. Shifted, F4 DELL deletes the current line. You can press the F4 key several times to insert or delete more than one line. It is not until you press RETURN that the results are shown.
UNDO F5 MARK	F5 MARK marks a line or lines for later use with another function. Shifted, F5 UNDO undoes previous changes to your file.
COPY F6 MOVE	F6 MOVE moves any marked lines or characters to the current line or character. Shifted, F6 COPY copies any marked lines or characters to the current line or character.
LEFT F7 HELP	F7 HELP accesses the FSE HELP file. Shifted, F7 LEFT moves your view of the file to the left.
RIGHT F8 QUIT	F8 QUIT exits FSE without making changes to your file permanent. Shifted, F8 RIGHT moves your view of the file to the right.

NOTES

- You must press RETURN after pressing a programmable function key.
- Function keys F1 through F14 are the top row of keyboard keys 1 through]. Each must be preceded by ESC . For example, to enter an F3, press:

  ESC + 3 + RETURN

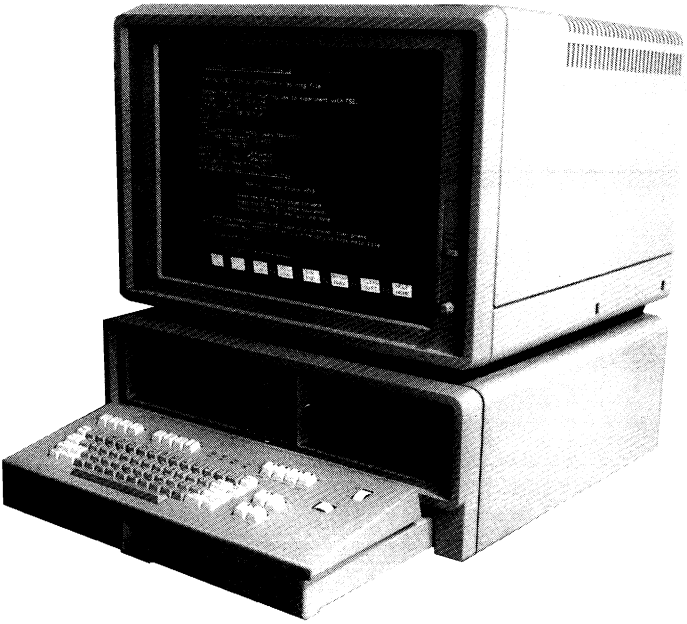
  To use the shifted functions of these keys, include the SHIFT key.

  ESC + SHIFT 3 + RETURN
- 0 and SHIFT 0 perform the same function, so in effect there is no shifted F10 function.

Miscellaneous Information

To specify tabs, use the soft tab character (the default character is \). Do not use the TAB key.

Tektronix 4115



Equivalent Keys

Viking	Equivalent on	
721 Key	Tektronix 4115	Function
FWD	F1	Advances screen one page.
BKW	SHIFT F1	Moves screen backward one page.
UP	F2	Moves current line to top of the screen.
DOWN	SHIFT F2	Moves current line to bottom of the screen.
HELP	SHIFT F8	Displays FSE HELP file.
BACK	BACK directive	Returns you to previously edited file.
⇧ INSRT	CTRL X	Places the terminal in a mode in which pressing any character key moves existing text to the right and inserts the new character. This mode is cancelled by pressing RTN.
INSRT	CTRL Z	Inserts a blank line allowing you to type in a new line of text.
⇧ DLETE	CTRL V	Deletes the current line.
DLETE	CTRL C	Deletes the current character.
⇧ CLEAR	SHIFT F7	Rewrites the entire screen.
HOME	F8	Positions the cursor at the FSE directive line, allowing you to enter FSE directives.
NEXT	RTN	Terminates an input line.
← 4	CTRL ← H	Moves cursor left.
↓ 2	CTRL ↓ J	Moves cursor down.
↑ 8	CTRL ↑ K	Moves cursor up.
→ 6	CTRL → L	Moves cursor right.

Default Programmable Function Keys for the Tektronix 4115

Key	Description
BKW F1 FWD	F1 FWD moves forward one page in the file. Shifted, F1 BKW moves backward one page in the file.
LINEDN F2 LINEUP	F2 LINEUP moves the current line to the top of the screen. Shifted, F2 LINEDN moves the current line to the bottom of the screen.
LAST F3 FIRST	F3 FIRST moves the cursor to the first line in the file. Shifted, F3 LAST moves the cursor to the last line in the file.
F4 UNDO	F4 UNDO cancels any marks you have set. If no marks were set, F4 UNDO cancels the most recent change to your file during the current editing session.
MRKCHR F5 MARK	F5 MARK marks a line for later use with another function. Shifted, F5 MRKCHR marks a character or characters for use with another function.
COPY F6 MOVE	F6 MOVE moves any marked lines or characters to the current line or character. Shifted, F6 COPY copies any marked lines or characters to the current line or character.
CLEAR F7 QUIT	F7 QUIT exits FSE without making the changes to your file permanent. Shifted, F7 CLEAR clears your screen.
HELP F8 HOME	F8 HOME positions the cursor on the FSE directive line. Shifted, F8 HELP displays the FSE HELP file on the lower half of your screen.

NOTES

- Unshifted function keys F9 through F12 are the following keyboard keys.

Function Key	Keyboard Key
F9	CTRL A
F10	CTRL S
F11	CTRL D
F12	CTRL F

- The shifted function keys F9 through F12 are the following keyboard keys.

Function Key	Keyboard Key
Shifted F9	CTRL Q
Shifted F10	CTRL W
Shifted F11	CTRL E
Shifted F12	CTRL R

Function Key Directive Strings E

This appendix contains the directive strings associated with the default settings of the programmable function keys.

The directive strings used by FSE are:

String†	Key Prompt
SKn/VNS/LABEL/FWD/	FWD
SKn/VPS/LABEL/BKW/	BKW
SKn/VN/LABEL/UP/	UP
SKn/VP/LABEL/DOWN/	DOWN
SKn/.I/LABEL/INSC/	INSC
SKn/.D/LABEL/DELC/	DELC
SKn/IBP1/LABEL/INSL/	INSL
SKn/D;PN/LABEL/DELL/	DELL
SKn/.END/LABEL/ENDLIN/	ENDLIN
SKn/SM/LABEL/MARK/	MARK
SKn/VH/LABEL/HOME/	HOME
SKn/SS/LABEL/CLEAR/	CLEAR
SKn/CMTP/LABEL/COPY/	COPY
SKn/MMTP/LABEL/MOVE/	MOVE
SKn/UNDO/LABEL/UNDO/	UNDO
SKn/QUIT/LABEL/QUIT/	QUIT
SKn/HELP/LABEL/HELP/	HELP
SKn/ /LABEL/ /	
SKn/SVO0/LABEL/LEFT/	LEFT
SKn/SVO&&C;.P+20/LABEL/RIGHT	RIGHT
SKn/PF/LABEL/FIRST/	FIRST
SKn/VL/LABEL/LAST/	LAST
SKn/IBP/LABEL/INSB/	INSB
SKn/DB/LABEL/DELB/	DELB
SKn/SVC80/LABEL/80COL/	80COL
SKn/SVC132/LABEL/132COL/	132COL
SKn"L/&&?/"LABEL"LOCATE"	LOCATE

† n can be any number from 1 through 16.

String†	Key Prompt
SKn/LN/LABEL/LOCNXT/	LOCNXT
SKn/V/LABEL/ MIDDLE/	MIDDLE
SKn/SMW/ LABEL/MRKCHR/	MRKCHR
SKn/.S/LABEL/SPLIT/	SPLIT
SKn/.J/LABEL/JOIN/	JOIN
SKn/.F/LABEL/PARA/	PARA
SKn/.C/LABEL/CENTER/	CENTER
SKn/SVC80;SVL34/LABEL/80COL/	80COL
SKn/SVC160;SVL64/LABEL/160COL/	160COL

Viking 721 Directive String Settings

The directive strings associated with the default settings for the Viking 721 terminal are:

String	Key Prompt
SK1/SM/LABEL/MARK/	MARK
SKS1/SMW/LABEL/MRKCHR	MRKCHR
SK2/MMTP/LABEL/MOVE/	MOVE
SKS2/CMTP/LABEL/COPY/	COPY
SK3/IBP/LABEL/INSB/	INSB
SKS3/DB/LABEL/DELB/	DELB
SK4/PF/LABEL/FIRST/	FIRST
SKS4/VL/LABEL/LAST/	LAST
SK5/UNDO/LABEL/UNDO/	UNDO
SK6/QUIT/LABEL/QUIT/	QUIT
SK7/"L/&&?/"LABEL"LOCATE"	LOCATE
SKS7/LN/LABEL/LOCNXT/	LOCNXT
SK8/SVC132/LABEL/132COL/	132COL
SKS8/SVC80/LABEL/80COL/	80COL
SK9/V/LABEL/ MIDDLE/	MIDDLE
SK10/.END/LABEL/ENDLIN/	ENDLIN
SK11/.S/LABEL/SPLIT/	SPLIT
SK12/.J/LABEL/JOIN/	JOIN
SK13/.F/LABEL/PARA/	PARA
SK14/CMTP/LABEL/COPY/	COPY
SK15/.C/LABEL/CENTER/	CENTER

† n can be any number from 1 through 16.

CDC 722 Directive String Settings

The directive strings associated with the default settings for the CDC 722 terminal are:

String	Key Prompt
SK1/VNS/LABEL/FWD/	FWD
SK2/VN/LABEL/LINEUP/	LINEUP
SK3/.I/LABEL/INSC/	INSC
SK4/IBP1/LABEL/INSL/	INSL
SK5/SM/LABEL/MARK/	MARK
SK6/MMTP/LABEL/MOVE/	MOVE
SK7/HELP/LABEL/HELP/	HELP
SK8/QUIT/LABEL/QUIT/	QUIT
SK9/.E/LABEL/ENDLIN/	ENDLIN
SKS1/VPS/LABEL/BKW/	BKW
SKS2/VP/LABEL/LINEDN/	LINEDN
SKS3/.D/LABEL/DELC/	DELC
SKS4/D;PN/LABEL/DELL/	DELL
SKS5/UNDO/LABEL/UNDO/	UNDO
SKS6/CMTP/LABEL/COPY/	COPY
SKS7/SVO0/LABEL/LEFT/	LEFT
SKS8/SVO&C;.P+20/LABEL/RIGHT/	RIGHT

DEC VT100 Directive String Settings

String	Key Prompt
SK	Clears all the keys
SK1/VNS/LABEL/FWD/	FWD
SK2/VN/LABEL/LINEUP/	LINEUP
SK3/.I/LABEL/INSC/	INSC
SK4/IBP1/LABEL/INSL/	INSL
SK5/SM/LABEL/MARK/	MARK
SK6/MMTP/LABEL/MOVE/	MOVE
SK7/HELP/LABEL/HELP/	HELP
SK8/QUIT/LABEL/QUIT/	QUIT
SK9/.END/LABEL/ENDLIN/	ENDLIN
SKS1/VPS/LABEL/BKW/	BKW
SKS2/VP/LABEL/LINEDN/	LINEDN
SKS3/.D/LABEL/DELC/	DELC
SKS4/D;PN/LABEL/DELL/	DELL
SKS5/UNDO/LABEL/UNDO/	UNDO
SKS6/CMTP/LABEL/COPY/	COPY
SKS7/VH/LABEL/HOME/	HOME
SKS8/SS/LABEL/CLEAR/	CLEAR

Zenith Z19 and Heathkit H19 Directive String Settings

String	Key Prompt
SK	Clears all the keys
SK1/VNS/LABEL/FWD/	FWD
SK2/VPS/LABEL/BKW/	BKW
SK3/VN/LABEL/LINEUP/	LINEUP
SK4/VP/LABEL/LINEDN/	LINEDN
SK5/.END/LABEL/ENDLIN/	ENDLIN
SK6/UNDO/LABEL/UNDO/	UNDO
SK7/QUIT/LABEL/QUIT/	QUIT
SK8/HELP/LABEL/HELP/	HELP
SKS1/SM/LABEL/MARK/	MARK
SKS2/SMW/LABEL/MRKCHR/	MRKCHR
SKS4/CMTP/LABEL/COPY/	COPY
SKS5/MMTP/LABEL/MOVE/	MOVE
SKS7/SVO0/LABEL/LEFT/	LEFT
SKS8/SVO&C;.P+20/LABEL/RIGHT/	RIGHT

Lear Siegler ADM3A Directive String Settings

String	Key Prompt
SK1/VNS/LABEL/FWD/	FWD
SK2/VN/LABEL/LINEUP/	LINEUP
SK3/.I/LABEL/INSC/	INSC
SK4/IBP1/LABEL/INSL/	INSL
SK5/SM/LABEL/MARK/	MARK
SK6/MMTP/LABEL/MOVE/	MOVE
SK7/HELP/LABEL/HELP/	HELP
SK8/QUIT/LABEL/QUIT/	QUIT
SK9/.END/LABEL/ENDLIN/	ENDLIN
SK10/SS/LABEL/CLEAR/	CLEAR
SK11/PF/LABEL/FIRST/	FIRST
SK12/IBP/LABEL/INSB/	INSB
SK13”L/&?/”LABEL”LOCATE”	LOCATE
SK14/.S/LABEL/SPLIT/	SPLIT
SK15/.F/LABEL/PARA/	PARA
SKS1/VPS/LABEL/BKW/	BKW
SKS2/VP/LABEL/LINEDN/	LINEDN
SKS3/.D/LABEL/DELC/	DELC
SKS4/D;PN/LABEL/DELL/	DELL
SKS5/UNDO/LABEL/UNDO/	UNDO
SKS6/CMTP/LABEL/COPY/	COPY
SKS7/SVO0/LABEL/LEFT/	LEFT
SKS8/SVO&C;.P+20/LABEL/RIGHT/	RIGHT
SKS9/SMW/LABEL/MRKCHR/	MRKCHR
SKS11/VL/LABEL/LAST/	LAST
SKS12/DB/LABEL/DELB/	DELB
SKS13/LN/LABEL/LOCNXT/	LOCNXT
SKS14/.J/LABEL/JOIN/	JOIN
SKS15/.C/LABEL/CENTER/	CENTER

Lear Siegler ADM5 Directive String Settings

String	Key Prompt
SK1/VNS/LABEL/FWD/	FWD
SK2/VN/LABEL/LINEUP/	LINEUP
SK3/.I/LABEL/INSC/	INSC
SK4/IBP1/LABEL/INSL/	INSL
SK5/SM/LABEL/MARK/	MARK
SK6/MMTP/LABEL/MOVE/	MOVE
SK7/HELP/LABEL/HELP/	HELP
SK8/QUIT/LABEL/QUIT/	QUIT
SK9/.END/LABEL/ENDLIN/	ENDLIN
SK10/SS/LABEL/CLEAR/	CLEAR
SK11/PF/LABEL/FIRST/	FIRST
SK12/IBP/LABEL/INSB/	INSB
SK13”L/&?/”LABEL”LOCATE”	LOCATE
SK14/.S/LABEL/SPLIT/	SPLIT
SKS1/VPS/LABEL/BKW/	BKW
SKS2/VP/LABEL/LINEDN/	LINEDN
SKS3/.D/LABEL/DELC/	DELC
SKS4/D;PN/LABEL/DELL/	DELL
SKS5/UNDO/LABEL/UNDO/	UNDO
SKS6/CMTP/LABEL/COPY/	COPY
SKS7/SVO0/LABEL/LEFT/	LEFT
SKS8/SVO&C;.P+20/LABEL/RIGHT/	RIGHT
SKS9/SMW/LABEL/MRKCHR/	MRKCHR
SKS11/VL/LABEL/LAST/	LAST
SKS12/DB/LABEL/DELB/	DELB
SKS13/LN/LABEL/LOCNXT/	LOCNXT
SKS14/.J/LABEL/JOIN/	JOIN

Tektronix 4115 Directive String Settings

String	Key Prompt
SK1/VNS/LABEL/FWD/	FWD
SK2/VN/LABEL/LINEUP/	LINEUP
SK3/PF/LABEL/FIRST/	FIRST
SK4/UNDO/LABEL/UNDO/	UNDO
SK5/SM/LABEL/MARK/	MARK
SK6/MMTP/LABEL/MOVE/	MOVE
SK7/QUIT/LABEL/QUIT/	QUIT
SK8/VH/LABEL/HOME/	HOME
SK9/SVC160;SVL64/LABEL/160COL/	160COL
SK10"L/&?/"LABEL"LOCATE"	LOCATE
SK11/.END/LABEL/ENDLIN/	ENDLIN
SK12/.S/LABEL/SPLIT/	SPLIT
SKS1/VPS/LABEL/BKW/	BKW
SKS2/VP/LABEL/LINEDN/	LINEDN
SKS3/VL/LABEL/LAST/	LAST
SKS5/SMW/LABEL/MRKCHR/	MRKCHR
SKS6/CMTP/LABEL/COPY/	COPY
SKS7/SS/LABEL/CLEAR/	CLEAR
SKS8/HELP/LABEL/HELP/	HELP
SKS9/SVC80;SVL34/LABEL/80COL/	80COL
SKS10/LN/LABEL/LOCNXT/	LOCNXT
SKS11/.F/LABEL/PARA/	PARA
SKS12/.J/LABEL/JOIN/	JOIN

Viking 721 Terminal Settings F

The Viking 721 terminal contains software setup switches, which should be set as shown here to ensure correct operation under FSE. These switches are software toggle switches. That is, you have a limited number of options (usually two) from which to choose. To change a switch setting, press the corresponding programmable function key. The first set of switches appears when you turn on your terminal. Set them as follows:

F1	F2	F3	F4	F5	F6	F7	F8
MODE 1 CYBER	MODE 2 PLATO	MODE 3 CP/M	MODE 4 DISK	MODE 5 4014	MODE 6	MODE 7 PACK	TRMNL TEST

Once you have set these as shown, press:

F1
MODE 1 CYBER

to select CYBER mode. At this point, the screen becomes blank except for the cursor.

Press:

(SETUP)

to display the next set of switches and change them, if necessary, to the following settings.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
Return	LINE ON	PRNTR OFF	MARGIN ON	ALERT SOFT	LOCK ALPHA	N PAD NORMAL	SCREEN ROLL	CYBER LARGE	MORE SELECT

Then, press:

F10
MORE SELECT

to set the next set of switches to the following.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
return	BACKGD DARK	CURSOR BLOCK	CURSOR SOLID	BAUD 1200	DUPLEX HALF	CHR/LN 80	LINES 30	XPARNT OFF	mode SELECT

When you have set all the switches, press:

F1
return

to remove the switch prompts from the screen. You are now ready to log in to NOS.

Refer to the *721-10/20/30 Hardware Reference Manual* for further information on these settings (ordering information is in the preface).

Index

Index

NOS 2 Full Screen Editor User's Guide 60460420 B

We would like your comments on this manual. While writing it, we made some assumptions about who would use it and how it would be used. Your comments will help us improve this manual. Please take a few minutes to reply.

Who Are You?

□ Manager
□ Systems Analyst or Programmer
□ Applications Programmer
□ Operator
□ Other ______________

How Do You Use This Manual?

□ As an Overview
□ To Learn the Product/System
□ For Comprehensive Reference
□ For Quick Look-up

Do You Also Have?

□ NOS 2 Reference Set, Vol 1
□ NOS 2 Reference Set, Vol 2
□ NOS 2 Reference Set, Vol 3

What programming languages do you use? ______________

Which are helpful to you? □ Directive Index □ Common Parameter Index □ Diagnostics (App B) □ Character Set □ Other: ______________

How Do You Like This Manual? Check those that apply.

Yes	Somewhat	No	
□	□	□	Is the manual easy to read (print size, page layout, and so on)?
□	□	□	Is it easy to understand?
□	□	□	Is the order of topics logical?
□	□	□	Are there enough examples?
□	□	□	Are the examples helpful? (□ Too simple □ Too complex)
□	□	□	Is the technical information accurate?
□	□	□	Can you easily find what you want?
□	□	□	Do the illustrations help you?
□	□	□	Does the manual tell you what you need to know about the topic?

Comments? If applicable, note page number and paragraph.

Continue on other side

Would you like a reply? □ Yes □ No

From:

Name ______________ Company ______________

Address ______________ Date ______________

______________ Phone No. ______________

____

Please send program listing and output if applicable to your comment.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

FOLD FOLD

Comments (continued from other side)