EDIT

PURPOSE

Edit a file.

DESCRIPTION

The EDIT command provides a full featured line editor. The editor is typically used to correct a macro file or a data file without exiting DATAPLOT. It is particularly useful on non-window systems where it is not possible to edit in a separate window.

This editor does not support screen mode editing at this time. Although a line editor is more primitive than a screen editor, it does provide maximum portability across terminals and operating systems.

Some summary documentation is provided in the NOTE section below (this information should be sufficient for making simple changes to a data or macro file). Enter QUIT to leave the editor and resume normal DATAPLOT use.

Most users already have a favorite editor. If your implementation supports the SYSTEM command, you may be able to use it to invoke your own editor. For example, on Unix implementations you can enter:

SYSTEM vi <file>   (or emacs or whatever editor you prefer)

Note that the SYSTEM command for the PC version will typically not work if this approach to editing is tried.

SYNTAX

EDIT <file name>
where <file name> is the name of the file to be edited.

EXAMPLES

EDIT PLOTCALIB.DP
EDIT PLOTCALIB.DAT

NOTE 1

When you are using EDIT, it is in one of 2 modes:

1. edit mode (the default mode); or
2. input mode.

When EDIT is in edit mode, then all of the usual editor commands (such as PRINT, CHANGE, DELETE, LOCATE, NEXT, COPY, MOVE, ADD, LIST, EXIT, etc.) are at the disposal of the analyst. The edit mode is the most common mode of operation. Most changing and updating of files is done in edit mode.

The advantage of edit mode is that all of the editor commands can be used to make the desired changes in the file. The disadvantage of edit mode is that extra keystrokes are required for the entry of large amounts of text.

When EDIT is in input mode, then “what you type” is “what goes into the file.” Input mode is the most common way of entering large amounts of text into a file.

The advantage of input mode is that large amounts of text can be efficiently entered. The disadvantage of input mode is that the analyst has no immediate access to editor commands (such as PRINT, DELETE, and so forth).

The only way to enter input mode from edit mode is to enter:

INPUT

The only way to enter edit mode from input mode is to enter:

EDIT

NOTE 2

The following is a brief summary of the EDIT sub-commands. More detail can be retrieved for many of these commands by entering HELP <command> while in the EDIT session (i.e., after entering the EDIT command but before entering QUIT). The examples sometimes use one of the abbreviated forms of the command. The full list of abbreviations is given in NOTE 3.

ABORT   Abort out of the editor without making any changes.
Example: ABORT

ABRR <file> Exit out of the current editing session without making any changes and resume the editor using the file specified by <file>. If no new file is specified, the editor goes into an infinite loop (this is a bug).

Example: ABRR JUNK.DAT

ADD <file> Insert the text file specified by <file> after the current line.

Example: ADD ABC.TEX

BOTTOM Go to the bottom (i.e., the last line + 1) of the file.

Example: B

CA /<old>/<new>/ <n> Change all occurrences of the string specified by <old> to the string specified by <new> on the next <n> lines starting with the current line. Be sure to include a space between the CA and the slash (/). If <n> is omitted, the change is applied to the current line.

Example: CA /KAT/CAT/
CA /KAT/CAT/ 20

CALL <file> Execute an editor macro (subprogram) stored in the file specified by <file>.

Example: CALL ABC.FED

CENTER <n> Center the next <n> lines, starting with the current line, at the column specified by the most recent SET CENTER command.

Example: SET CENTER 25
CENTER

CHANGE /<old>/<new>/ <n> Change the first occurrence of the string specified by <old> to the string specified by <new> on the next <n> lines starting with the current line. Be sure to include a space between the C and the slash (/). If <n> is omitted, the change is applied to the current line.

Example: C /KAT/CAT/
C /KAT/CAT/ 20

CLOSE Close the printer file on IBM-PC implementations and the file PRINT.DAT on other implementations. This command is used in conjunction with the OPEN command.

Example: CLOSE

COPY <file> <n1> <n2> Copy the lines between <n1> and <n2> to the file specified by <file>. If <file> is omitted, the text is copied to the file EDCOMM.TEX (the name may vary on some systems, e.g. it is edcomm.tex on Unix systems). If <n1> and <n2> are omitted, then the lines marked by a prior SET COPY command or by the SC1 and SC2 commands are copied. An alternate form of this command is COPY <n1> <n2> <n3>. In this syntax, the lines between <n1> and <n2> are copied after line <n3>.

Example: SET COPY 20 30
COPY ABC.TEX
COPY ABC.TEX 20 30
COPY 10 20 50

CTL <string> Apply the most recently specified change (see the CHANGE command above) from the current line until the next line that contains the string specified by <string>.

Example:C /KAT/CAT
CTL CAT

CUT <string> Trim all text on the current line from <string> to the end of the line.

Example: CUT CAT

DCOPY <file> This command is similar to the COPY command (see above). The distinction is that the copied lines are deleted after being copied.

Example: SET COPY 20 30
DCOPY ABC TEX

DELETE <n1> <n2> Delete all the lines between <n1> and <n2>. If <n2> is omitted, delete the next <n1> lines of text starting with the current line. If both <n1> and <n2> are omitted, the current line is deleted.

Example: D
D 10
D 5 14

DHOLD <n> Delete the current line, but store it in the internal buffer <n> for later use by the DUP command. If <n> is omitted, buffer 1 is used. Up to 10 buffers can be specified.

Example: DHOLD
DHOLD 10

DI <string>
Delete the current line and replace it with the text specified in <string>.
Example: DI Enter this on the next line

DUP <n1> <n2>
Duplicate held lines from a prior HOLD or DHOLD command. If <n1> and <n2> are both omitted, then the first buffer is duplicated after the current line. If only <n1> is specified, then buffer <n1> is duplicated after the current line. If both <n1> and <n2> are specified, then buffer <n1> is duplicated <n2> times after the current line. If <n1> is an empty buffer, then a blank line is copied.
Example: . Copy line 22 to after line 46

```
22
HOLD
46
DUP
. Copy line 22 to buffer 2 and copy it after line 46
22
HOLD 2
46
DUP 2
. Copy line 22 to buffer 2 and copy it 10 times after line 46
22
HOLD 2
46
DUP 2 10
```

DTL <string>
Delete all lines from the current line until the first line containing <string>.
Example: DTL CAT

EDIT
Switch from input mode to edit mode.
Example: EDIT

ERASE
Clear the terminal screen. This command is only active for a few specific terminal types.
Example: ERASE

EXECUTE
Execute a held line as an editor command. This is typically only used when editing an editor macro file.
Example:18

```
HOLD
X
```

EXTEND <string>
Append <string> onto the end of the current line.
Example: EXT SUBSET

EXTA <string>
Append <string> onto the end of lines from the current line to the end of the file.
Example: EXTA SUBSET X = 2

EXIT
Exit out of the current editing session and make the changes permanent.
Example: EXIT

EXRR <file>
Exit out of the current editing session, making the changes permanent, and resume the editor using the file specified by <file>. In versions prior to 95/2, the editor goes into an infinite loop if no file is specified.
Example: EXRR ABC.TEX

FIND <string>
Search for the next occurrence of <string> starting with the next line and the match starting in column 1. The SET FIND command can be used to specify the columns for the FIND command (in which case the match starts in first column specified in SET FIND).
Example: FIND RESIDUALS

FIRST
Go to the first line of the file. Versions prior to 95/2 have a bug and interpret FIRST as FIND.
Example: FIRST

FORMAT
This command is currently inactive.

GO <n1>
Go to line <n1>. The word GO can be omitted (i.e., simply list a line number). If <n1> is omitted, go to the line specified by the SET MARK command. If no SET MARK was previously specified, go to the top of the file.
Example: GO 25

```
25
```

HELP Display on-line help information.
Example: HELP
HELP PRINT
HELP CHANGE

HOLD <n> Store the current line in the internal buffer <n> for later use by the DUP command. If <n> is omitted, buffer 1 is used. Up to 10 buffers can be specified.
Example: HOLD
HOLD 2
HOLD 10

IF CHANGE <YES/NO> <command> This command is typically used only in edit macro files and is preceeded by a CHANGE command. If the most recent CHANGE command completed successfully, then execute <command>. If it did not complete successfully, skip <command>.
Example: IF CHANGE YES EXIT
IF CHANGE NO ABORT

IF LOCATE <YES/NO> <command> This command is typically used only in edit macro files and is preceeded by a LOCATE command. If the most recent LOCATE command completed successfully, then execute <command>. If it did not complete successfully, skip <command>.
Example: IF LOCATE YES C /-999/0/
IF LOCATE NO EXIT

INDENT <n> Indent the next <n> lines, starting with the current line, to the column specified by the SET INDENT command..
Example: SET INDENT 3
INDENT
INDENT 10

INPUT Switch from edit mode to input mode. Use the EDIT command to leave input mode.
Example: INPUT

INSERT <string> Insert <string> on a new line after the current line.
Example: I PLOT Y X

LA <string> Locate and print all lines between the current line and the last line of the file containing <string>.
Example: LA PLOT

LABL Locate and print all blank lines that occur between the current line and the end of the file.
Example: LABL

LAST Go to the last line of the file.
Example: LAST

LB <string> Locate and print the first line containing <string>. Begin with the current line and work back to the beginning of the file.
Example: LB PLOT

LBA <string> Locate and print all lines containing <string>. Begin with the current line and work back to the beginning of the file.
Example: LBA PLOT

LBL Locate and print the first blank line that occurs between the current line and the end of the file.
Example: LBL

LC <string> <file> Locate the next line containing <string> and call the edit macro specified by <file>. The text in <string> should not contain any spaces. There is currently a bug if the file specified by <file> cannot be found (all succeeding commands return a message saying the file cannot be found).
Example: LC PLOT CHANGE.FED

LDBL Locate and delete all blank lines that occur betwen the current line and the end of the file.
Example: LDBL

LDEL <string> Locate and delete all lines containing <string> that occur between the current line and the end of the file.
Example: LDEL PLOT

LI <string1> <string2> Locate all lines between the current line and the end of the file containing <string1> and insert <string2> on the line after the located line. <string1> should not contain any spaces.
Example: LI PLOT PAUSE

LIB <string1> <string2> Locate all lines between the current line and the end of the file containing <string1> and insert <string2> on the line before the located line. <string1> should not contain any spaces. <string2> contains all the text from the first non-blank character after <string1> to the last non-blank character on the line.
Example: LIB PLOT TITLE PLOT SAMPLE ^K

LIST <file> List an external file.
Example: LIST ABC.TEX

LJNF <string> This command is only used from within an edit macro file. Search the current line and all lines thereafter until the end of the file for the first occurrence of <string>. If <string> is not found, skip all lines in the edit macro file (invoked with the CALL command) until the line containing END OF LJNF is located.
Example: LJNF PLOT

LKIL <string> Delete all lines between the current line and the last line of the file that do not contain <string>.
Example: LKIL PLOT

LLD <string> Locate back-to-back lines containing <string> and delete the second line.
Example: LLD PLOT

LO <string> Locate the next line, starting with the current line, containing <string>.
Example: L PLOT

LOOP <n> <com> Execute <com><n> times.
Example: LOOP 3 LC PLOT MACRO.FED

LP <string> Locate all lines, starting with the line after the current line, until the end of the file containing <string>. Instead of printing the located line, print the located line plus <n> where <n> is specified by the SET LP OFFSET command.
Example: SET LP OFFSET 1
LP PLOT

LPER Locate and print the next line after the current line that contains a period and nothing else except spaces.
Example: LPER

LS <string> Locate all lines between the current line and the end of the file that contain <string> and replace <string> with a sequence number.
Example: LS XXX

LSRJ <string> Locate all lines between the current line and the end of the file that contain <string> and replace <string> with a right justified sequence number.
Example: LSRJ XXX

NEAT Make a paragraph neat.
Example: SET NEAT 1 50
NEAT

NEWS Print the contents of the editor news file. As this editor is fairly mature, updates should be fairly rare.
Example: NEWS

NEXT <n> Go to the current line plus <n>. If <n> is omitted, go to the next line. Entering an empty carriage return is equivalent to NEXT with <n> omitted.
Example: N
N 10

OPEN Open the printer file on IBM-PC implementations and the file PRINT.DAT on other implementations. This command is used in conjunction with the CLOSE command. This syntax is used when you want to append the output from more than one PRINT command to PRINT.DAT. That is, enter the OPEN command, enter various PRINT commands, and then enter CLOSE. If this syntax is not used, then each PRINT command rewinds PRINT.DAT and overwrites the file.
Example: OPEN
PN 10
GO 90
PN 10
CLOSE

PA Print all lines from the current line to the end of the file.
Example: PA

PBL
Print a blank line. This command is typically only used in edit macros.
Example: PBL

PDL
Print a dashed line. This command is typically only used in edit macros.
Example: PDL

PEL
Print a line with “ERROR OCCURED” message. This command is typically only used in edit macros.
Example: PEL

PN <n>
Print the next <n> lines starting with the line after the current line. If <n> is omitted, print the next line.
Example: PN 20
PN

PP
Print a page of text. The number of lines for a page can be set with the SET PP LINES command. The syntax PP2 sends the output to the local printer on IBM-PC implementations and to the file PRINT.DAT on all other systems.
Example: PP

PPAR
This command is currently inactive. It is reserved for printing a paragraph.

PRINT <n1> <n2>
Print lines <n1> through <n2> inclusive. If <n2> is omitted, print the next <n1> lines starting with the current line. If both <n1> and <n2> are omitted, print the current line. The syntax P2 sends the output to the local printer on IBM-PC implementations and to the file PRINT.DAT on all other systems. Be aware that on non-PC systems, the file PRINT.DAT is over-written for each PRINT command. If you need to append several PRINT commands, use the OPEN and CLOSE syntax instead.
Examples: PRINT 5 14
PRINT 10
PRINT

PTL <string>
Print all lines between the current line and the end of the file until a line that contains <string> is located.
Example: PTL PLOT

RESTORE
Retrieve the settings of a previous editing session (which was saved with the SAVE command).
Example: RESTORE

SAVE
Save the settings of the current editing session (which can be retrieved with the RESTORE command).
Example: SAVE

SCALE <n>
Print a header line (i.e., 123456789.123456789. etc.) for <n> columns.
Example: SCALE 60

SET BEGIN <n>
Set the first line for a subsequent COPY command. It is paired with a SET END command.
Example: SET BEGIN 25
SET END 52
COPY JUNK.DAT

SET CENTER <n>
Mark the column for the CENTER command. The line will be centered about column <n>.
Example: SET CENTER 25

SET CHAN <n1> <n2>
Subsequent CHANGE commands are restricted to the columns between <n1> and <n2> inclusive.
Example: SET CHANGE 7 20

SET COPY <n1> <n2>
Set the first and last lines for a subsequent COPY command. It is equivalent to a SET BEGIN and SET END pair.
Example: SET COPY 70 90

SET ECHO <ON/OFF>
Specify whether the editor prints the command enclosed in a box of asterisks (ON) or not (OFF).
Example: SET ECHO ON
SET ECHO OFF

SET END
Set the last line for a subsequent COPY command. It is paired with a SET BEGIN command.
Example: SET BEGIN 25
SET END 52
COPY JUNK.DAT

SET FEEDBACK <ON/OFF>
Specify whether the editor prints feedback messages (ON) or not (OFF).
Example: SET FEEDBACK ON
SET FEEDBACK OFF

SET FIND <n1> <n2> Set the begininng and ending columns for a subsequent FIND command.
Example: SET FIND 4 22
FIND PLOT

SET IBUGED <ON/OFF> Specify whether certain debugging information is printed (ON) or not (OFF). In addition, the additional following bug switches can be set: IBUGE2, IBUGE3, IBUGTY, IBUGFI, IBUGT1, IBUGT2, IBUGWR, and IBUGMA. Debugging for a specific routine can be turned on by entering SET ISUBRO <name> where <name> is the last 4 characters of the desired subroutine. This command is typically used for debugging purposes, so users do not normally set these switches ON.
Example: SET IBUGED ON
SET IBUGED OFF

SET INDENT <n> Set the number of columns for the INDENT command. INDENT will indent lines by <n> columns.
Example: SET INDENT 5

SET LOCA <n1> <n2> Mark the columns for the LOCATE command.
Example: SET LOCATE 5 45
L SUBSET

SET LP OFFSET <n> Specify that the LP command start printing <n> lines past the located line.
Example: SET LP OFFSET 1

SET MARK <n> Set a mark at line <n>. By default, the GO command with no argument goes to the top of the file. If the SET MARK command is entered, subsequent GO commands without arguments go to the marked line.
Example: SET MARK 34

SET MASK <char> Set the mask character to <char>.
Example: SET MASK #

SET NEAT <n1> <n2> Mark the columns for the NEAT command.
Example: SET NEAT 1 50

SET PP LINES <n> Specify that the PP command print <n> lines. If <n> is omitted, the default of 50 lines is reset.
Example: SET PP LINES 25

SET PP OFFSET <n> Specify that the PP command start printing <n> lines past the current line.
Example: SET PP OFFSET 5

SET PRINT <ON/OFF>Specify whether the print command sends output to the screen (OFF) or to a printer (ON). SET PRINT ON is equivalent to using the P2 or the PP2 commands (see PRINT and PP commands for details). This command actually only goes to the printer for the IBM-PC implementation (it goes to the local printer identified by PRN). For other systems, the PRINT output goes to the file PRINT.DAT.
Example: SET PRINTER ON
SET PRINTER OFF

SET PROMPT <ON/OFF> Specify whether the editor prints the prompt character “>” to indicate it is waiting for the next command (ON) or not (OFF).
Example: SET PROMPT ON
SET PROMPT OFF

SET SEQUENCE <n> Set the beginning sequence number for the LS and LSRJ commands.
Example: SET SEQU 10

SET SHIFT <n> Specify the number of columns to shift on the SHIFT command. Positive numbers shift the line right while negative numbers shift the line left.
Example: SET SHIFT 55

SET TRUNCATE <n> Specifies that the TRUNCATE command trucates lines at column <n>.
Example: SET TRUNCATE 55
TRUCATE 10

SC1 Mark the first line for the COPY command. Used with SC2, this is equivalent to SET COPY or SET BEGIN and SET END.
Example: SC1

SC2 Mark the last line for the COPY command. Used with SC1, this is equivalent to SET COPY or SET BEGIN and SET END.
Example: SC2

SHIFT
Shift text left or right on lines.
Example: SET SHIFT 5
SHIFT
SHIFT 20

SHOW
Display settings of switches and limits.
Example: SHOW

SPLIT
This command is currently inactive. It is reserved for splitting the remainder of a line onto the next line.

STAT
Display the settings of switches and limits.
Example: STAT

TOP
Go to the top (i.e., line 0) of the file.
Example: T

TRUNCATE <n>
Truncate the next <n> lines, starting with the current line, at the column specified by the SET TRUNCATE command. If <n> is omitted, the current line is truncated.
Example: SET TRUN 55
TRUNCATE
TRUNCATE 300

UNDO
Undo all changes from the most recent TOP.
Example: UNDO

UP <n>
Go up <n> lines from the current line. If <n> is omitted, go to the previous line.
Example: UP
UP 10

. (period)
Signifies a comment line (primarily used for edit macros). Be sure to include a space after the period.
Example: . The next line is a global edit

<n>
Go to line <n>.
Example: 8

/
Print the next 20 lines.
Example: /

\
Print the previous 20 lines.
Example: \

NOTE 3

The following is a list of the accepted abbreviations (or synonyms) for the EDIT subcommands.

ABORT	--AB	ABO	ABO+anything		
ABRR	--ABR				
ADD	--AD				
BOTTOM	--B	B+anything			
CALL	--CAL	CAL+anything			
CENTER	--CE	CE+anything			
CHANGE	--C	CH	CH+anything		
CA	--CG				
COPY	--CO	COP			
CTL	--				
CUT	--CU				
DCOPY	--DCOP	MOVE	M	M+anything	
DELETE	--D	DE	DE+anything		
DH	--HD	DH+anything			
DI	--				
DUP	--DU	DU+anything			
DTL	--				
EDIT	--EDIT				
ERASE	--ER	ERA	ERAS		
EXECUTE	--EXE	EXEC	EXEC+anything		X
EXTEND	--EXT	EXT+anything except A			
EXTA	--EA				

EXIT	--E	EX	EXI	END, HALT, QUIT, STOP, BYE
EXRR	--			
FIND	--F	FI	FI+anything	
FIRST	--FIR	FIRS		
FORMAT	--FO	FO+anything		
GO	--G			
HELP	--HE	HE+anything		
HOLD	--H	HO	HO+anything	
IF CHANGE	--IF CHAN			
IF LOCATE	--IF LOCA			
INDENT	--IND	IND+anything		
INPUT	--INP+anything			
INSERT	--I	IN	INS	INS+anything
LA	--			
LABL	--			
LAST	--LAS	LAS+anything		
LB	--			
LBA	--			
LBL	--LOBL			
LC	--LC			
LDBL	--			
LDEL	--			
LI	--			
LIB	--			
LIST	--LIS			
LJNF	--			
LKIL	--			
LO	--L	LOC	LOCA	LOCATE
LOOP	--			
LP	--			
LPER	--			
LS	--			
LSRJ	--			
NEAT	--NEA	NEA+anything		
NEWS	--			
NEXT	--N	N+anything		
PA	--			
PBL	--			
PDL	--			
PEL	--			
PN	--			
PP	--			
PPAR	--			
PRINT	--P	PR	PR+anything	
PTL	--PL			
RESTORE	--RES	REST		
SAVE	--SA	SAV		
SCALE	--SC	SCA		
SC1	--			
SC2	--			
SET	--S	SE		
SET MARK	--SM			
SHIFT	--SHI	SHI+anything		
SHOW	--SH	SHO		
SPLIT	--SP	SP+anything		
TOP	--T	TO		
TRUNCATE	--TR	TRU	TRU+anything	

UNDO	--		
UP	--U		
/	--?	‘	“
\	--|		

NOTE 4

The following are the defaults and limits for EDIT.

Maximum number of characters per line	=	240
Maximum number of lines in workspace	=	25,000
Maximum number of characters in workspace	=	500,000
Column limits for CHANGE command	=	1 to 132
Column limits for PRINT command	=	1 to 132
Mask character	=	*
Feedback	=	ON
Auto line numbering	=	ON
Trace	=	OFF
Justification	=	LEFT
Terminal	=	Any alphanumeric terminal
Terminal Rows	=	Any number of rows
Terminal Columns	=	60+ characters (else wrap-around)

NOTE 5

The file EDSYST.TEX in the DATAPLOT reference file directory is a system start-up file for the EDIT command. The local DATAPLOT implementor can use this file to define any site-wide defaults. The file EDLOGI.TEX in the current directory is a user defined start-up file for the EDIT command. On the PC implementation, EDLOGI.TEX is also in the DATAPLOT reference file directory. The creation of an EDLOGI.TEX is totally optional on the part of the user.

DEFAULT

None

SYNONYMS

FED

RELATED COMMANDS

CREATE	=	Starts copying commands to macro file.
LIST	=	Lists the contents of a file.
SEARCH	=	Search a file for a string.

APPLICATIONS

Editing files

IMPLEMENTATION DATE

92/6

PROGRAM

```
LIST MYFIT.DP
EDIT MYFIT.DP
LO FIT Y X1 X2 X3
C /X1 X2 X3/X1 X2/
EXIT
CALL MYFIT.DP
```