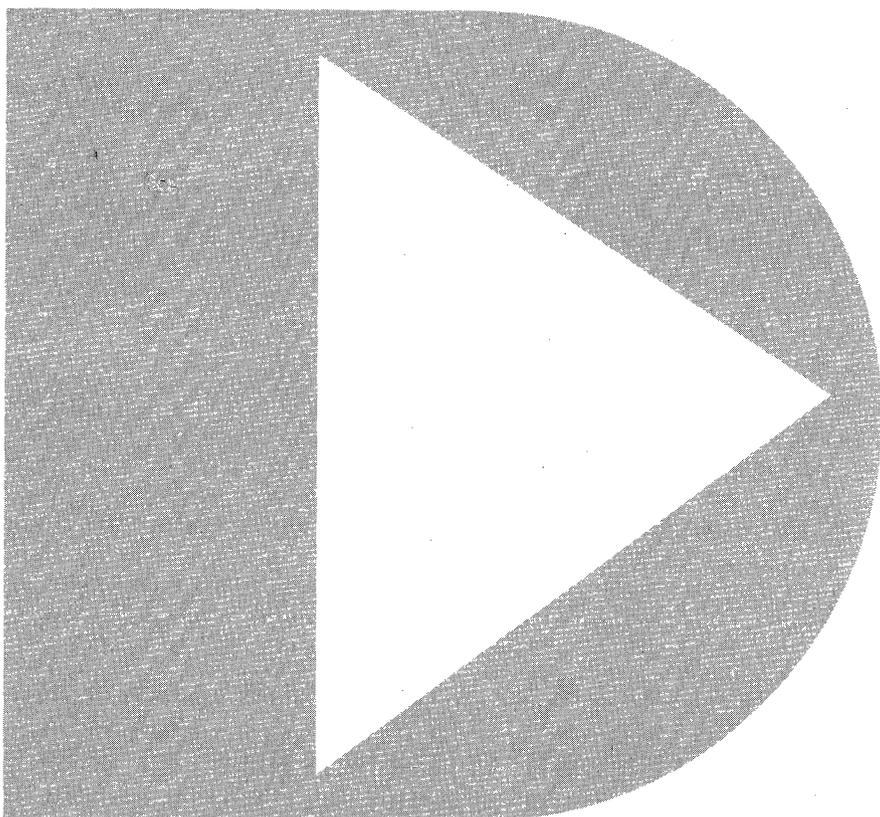


Program Users Guide

Datapoint



CASSETTE AND DOS EDITOR

USERS GUIDE

July 12, 1973

TABLE OF CONTENTS

	Page
1. INTRODUCTION	
2. OPERATION	
2.1 CASSETTE INITIALIZATION	1-1
2.1.1 Tape Files	2-1
2.2 DOS INITIALIZATION	2-1
2.2.1 Disk Files	2-1
2.3 PARAMETER LIST	2-2
2.3.1 Margin Bell	2-2
2.3.2 Tab Key Character	2-2
2.3.3 Mode	2-2
2.3.4 Update	2-2
2.3.5 Tape Format	2-3
2.4 EXAMPLES	2-3
2.4.1 DOS Examples	2-3
2.4.2 Cassette Example	2-4
2.5 DATA ENTRY	2-4
2.6 DATA RETRIEVAL	2-5
2.7 EDITOR COMMAND FORMAT	2-5
3. COMMANDS	
4. MODIFICATION COMMANDS	
4.1 DELETE COMMAND	4-1
4.2 MODIFY COMMANDS	4-1
4.2.1 Line Modification	4-1
4.2.2 Field Modification	4-2
5. FILE SEARCH COMMANDS	
6. MISCELLANEOUS COMMANDS	
7. RECOVERY PROCEDURES	
7.1 CASSETTE RECOVERY	4-1
7.1.1 Parity Errors During Editing	7-1
7.1.2 Parity Errors During Copy Back	7-1
7.1.3 Power Failure or Abort	7-1
7.2 DOS RECOVERY	
7.2.1 Bypassing Errors or End of File	7-2

8. GLOSSARY

9. COMMAND LIST

1. INTRODUCTION

The General Purpose Editor (GEDIT 2.1 or DOSGEDIT 2.1) enables the user to create and to update source data files on the disk or cassette. The editor, through the use of initialization parameters, will enable the user to create files in a variety of formats: text files, assembler code files, DATABUS source code files, or user designed data files.

A GLOSSARY of the many terms and phrases used throughout this manual is provided in section 8 and a list of commands and brief definitions is provided in section 9.

Throughout the manual these conventions will be used:

Square brackets - [] - indicate optional fields.

Pointed brackets - < > - indicate required fields.

Caution: Although virtually any Datapoint format file may be "edited", files structured with respect to physical records or those containing strings longer than 79 characters may find that organization collapsed as the editor compresses the file into sequential format. In such cases the editor should not be used.

2. OPERATION

2.1 CASSETTE INITIALIZATION

Place the GEDIT (LGO) tape in the rear deck of the Datapoint 2200 and press the RESTART key. When the editor is ready, the message:

```
GEDIT 2.1
OLD,NEW,DUP;PARAMETERS
?
```

will be displayed. The operator must type in one of the three mentioned tape designators (OLD or NEW or DUP).

2.1.1 Tape Designation

Single letter tape designation (O, N or D) are sufficient.

'NEW' tape means that the data tape in the front deck contains nothing (or nothing of value since it will be written over). As the operator enters text it will be written directly onto the front tape.

'OLD' tape means that the data tape in the front deck contains the information (previously written by GEDIT) to be edited. As an old tape is read and, as changes are made and new information added, the text is transferred to a 'scratch' file on the end of the GEDIT tape. This scratch area will later be transferred back to the data tape in order to update its contents.

If the GEDIT tape has previously been used to edit an 'old' tape, its scratch area will contain a copy of that tape. To retrieve this copy, specify the 'DUP' tape designation. A blank tape should be in the front deck to receive the copied data file.

A SEMI-COLON (;) separates the parameter list from the tape designator. If no parameter list is provided, Assembler mode is assumed (see Glossary).

2.2 DOS INITIALIZATION

The DOSGEDIT program, catalogued EDIT/CMD on your disk, is parameterized as follows:

```
EDIT <f1>[,f2][;parameter list]
```

2.2.1 Files

The source file <f1> is assumed to have an extension of 'TXT' if none is provided. If there is no file of the specified name, one will be opened. If no scratch file <f2> is specified, a file 'SCRATCH/TXT' will be used.

2.3 PARAMETER LIST

A parameter list, indicated by the SEMI-COLON (;) following the file specification (or tape designation in the cassette version), may be included. That list may include up to seven parameters which are order independent. The possible parameters are:

[;[margin][tab key][mode][shift][line][update][format]

If no parameter list is provided, Assembler mode with a margin at 75 and SPACE bar for tabbing is assumed.

2.3.1 Margin Bell

A number in the parameter list will be taken to be the margin designator; this causes the margin 'bell' to ring at the designated margin. (Text may always be input up to column 79 regardless of the margin setting.)

For Example ;30 will cause the bell to ring in column 30.

2.3.2 Tab Key Character

A tab key character encountered in the parameter list, i.e., a non-alpha, non-numeric, non-colon, will replace the assumed tab key character. (SPACE in Assembler, DATABUS and Comment mode, SEMI-COLON in Text mode.)

For Example ;^ will cause the tab key caret (^) to replace the assumed character.

2.3.3 Mode

A new set of assumptions will be used if one of the 'mode' parameters is set. If no mode is listed or 'A' is typed, Assembler mode will be used. DATABUS (D) mode simply changes the tab stops. Comment mode (C) changes the nature of the DELETE and SCRATCH commands to facilitate adding or changing comments on assembly code files.

Text mode (T) sets no tabstops, does no shift inversion and enables the word wrap around feature (see the glossary). To activate line truncation instead of word wrap around in Text mode, enter 'L' in the parameter list. To enable shift

key inversion (see glossary) in Text mode, enter the parameter 'S' in the list. Text mode is especially useful for generating SCRIBE input files.

See the glossary for complete definitions of the various modes.

2.3.4 Update

During editing, the source file (or input tape) is transferred into the scratch file as the text is updated. The physical source file may be used as the scratch file as the edit proceeds. When the edit is terminated, the physical source file is updated.

To inhibit source file update, the 'ONE-PASS' option 'O' may be set in the parameter list. A flag is set which prevent writing on the physical source file. Then, at the completion of the edit, the scratch file will contain the updated information and the source file will be unchanged.

In the cassette version, if the one-pass option is set, the message:

BLANK TAPE IN REAR DECK?

will be displayed. The operator should remove the GEDIT tape and place a scratch tape in the rear deck. When ready, press the ENTER key. When the edit is completed (after the :E or :E/ command has been executed), the message:

DONE - RELOAD GEDIT TAPE

will appear. At this time the cassette in the rear deck contains the updated file and the front data tape is unchanged. Replace the GEDIT tape and press the ENTER key. This will cause the editor to restart.

2.3.5 Tape Format (Cassette only)

Use the 'W' parameter to set 'write-edit' tape format. This will cause the tape records to be exactly 80 data characters with trailing blanks supplied and no space compression.

If no parameter is supplied, a standard, blocked, compressed, data format is assumed. This format is compatible with all Datapoint terminal emulators.

2.4 EXAMPLES

2.4.1 DOS Examples

To perform standard Assembler code editing, enter the command:

```
EDIT <source>
```

To edit a file for input to the text processor SCRIBE, the command:

```
EDIT <source>;T
```

is adequate. To change the margin bell to ring at column 35 (e.g. for labels) the command:

```
EDIT <source>;35T
```

would set the bell and use the Text mode assumptions. Note that the parameters are order independent; therefore, the command: EDIT <source>;T35 would achieve the same results.

To generate a second, slightly different file (without updating the original file), the command:

```
EDIT <source>,<new file>;OT
```

protects the source. If the file is assembler code instead of text, simply omit the 'T'; if DATABUS, replace 'T' by 'D'.

A second file, with the same name as <f1> but with a different extension, may be used by entering:

```
EDIT <f1>,</extension>
```

Once the initial command (and parameter list) has been entered. The message GEDIT 2.1 will appear on the screen. This message will be rolled up and the screen cleared with the cursor left on the 'command line'. From this position data may be entered, lines may be fetched from the source file, or editor commands may be executed.

2.4.2 Cassette Examples

To generate a new text file (e.g. for input to SCRIBE), answer:

```
N;T
```

To update a previously created assembler code file, answer simply 'O' since Assembler mode is assumed.

To preserve the input tape, use the command:

```
O;O or O;OD or O;TO
```

which will create an updated tape on the rear deck without disturbing the original tape.

To generate a data tape in write-edit format the answer:

N;TSWL

might apply. This answer would enable text mode (T) (i.e., no tab stops) with no shift key inversion (S) or word wrap around (L) and the output tape would be in write-edit format (W) containing 80 character records.

2.5 DATA ENTRY

To enter text, simply type on the bottom line, when the ENTER key is pressed the screen rolls up one line. The command line is once again blank and the cursor is at the beginning of the command line, ready to accept more input.

If wrap around is in effect, when a SPACE is typed within the last 10 columns of the line or typing proceeds past the end of the line, the same action occurs. If a non-space character is typed into the last column, the last word on the line is removed and, after the screen is rolled up, that word is placed on the command line, where data entry may proceed.

When typing on a 'screen line' (as the result of a command), the ENTER key causes the cursor to return to the command line. To continue data entry at the same screen area, the Pseudo-ENTER key may be used. This key (DEL shifted) causes (in all but command mode), a new blank line to be inserted at that point on the screen so that data entry may proceed.

If word wrap around is enabled, and data is being entered on a screen line, a new line will automatically be inserted at that point when, as on the bottom line, a space is entered within the last 10 columns or a character is typed past column 79.

The BACKSPACE key erases the last character and moves the cursor back one position. The CANCEL key erases the line back to the previous tabstop (in text mode this would erase the entire line).

Typing the tab key character causes the cursor to move to the next tab stop to the right. If there are no tab stops to the right of the cursor, the tab key character is accepted as a normal data character.

2.6 DATA RETRIEVAL

To fetch data from the source file, press the KEYBOARD and DISPLAY keys simultaneously. As long as the two keys are depressed, data will be fetched, displayed on the command line and rolled up on the screen. If end of file is reached, no more data is fetched and the machine beeps.

To fetch a single line, the DEL key may be pressed (in the first column of the command line). Using this key insures that only one input line will be fetched.

2.7 EDITOR COMMAND FORMAT

The text appearing on the eleven screen lines (i.e. the lines above the command line) may be edited using a set of 'commands'. A 'pointer' (>) in the left hand column of the screen indicates the line which the command will affect.

To move the pointer up, press the DISPLAY key. The KEYBOARD key moves the pointer down. The pointer wraps around from the top to the bottom and vice versa.

Commands allow the user to delete a single line (:D) or part of the screen (:SC and :SB), insert (:I) a new line between the current lines on the screen and to modify parts of a line by replacing text or inserting new text. Commands are also available to search the file for specific text (:F and :L) or for the end of the file (:EO or :E*).

An editor command is always preceded by a COLON (:). To enter a command, type in the first column of the command line a colon and the appropriate command character and any necessary parameters. The command is always typed with the machine in lower case; thus, with shift inversion on (as in Assembler, Databus and Comment modes), the command character will appear upper case; while with shift inversion off (as in Text mode), it will appear lower case.

3. COMMANDS

The following commands are a subset of those available. The user can get started without worrying about complex command forms. Remember that the 'pointer' on the screen indicates the line affected by the command.

:D - DELETE - in all but Comment mode this command deletes the entire pointed line. (In Comment mode, only the comment field is deleted. The CANCEL key may however be used to delete the preceding fields in the line.)

The cursor is left on the now null line where new text may be entered. If no replacement text is needed, be sure to press the ENTER key in the first column of the pointed line, since trailing blanks will not generally be truncated.

Pseudo-ENTER may be used to generate additional lines at this area of the screen. Word wrap around, if in effect, will apply to text entered on a deleted line. Pressing the ENTER key will return the cursor to the command line.

See the section on modification for more information.

:E* - EOF without display - searches for the end of the file and, when it is reached, displays the last screen of text. The search may be aborted by pressing the KEYBOARD and DISPLAY keys simultaneously.

:EO - EOF with display - causes the data to be displayed on the screen continuously until end of file is reached. The display may be stopped at any time by pressing the KEYBOARD and DISPLAY keys.

:F <old text> - FIND match - the screen is cleared and the input file is searched for a line starting with the specified <old text>. Leading spaces in the input lines will be ignored and should not be entered as part of <old text> (note that this command should be typed exactly :F<SPACE><old text>).

A FIND will wrap entirely around the file (or up to the end of file if the one-pass option is set). If the requested text is not found, the last line on the screen when the FIND was executed will be displayed. A FIND may be stopped by pressing the KEYBOARD and DISPLAY keys.

:I - INSERT - Perform a line insert at the pointed line. This command causes the lines from the top of the screen to the pointed line, inclusive, to be rolled up and a blank line to be inserted. The cursor is left at the beginning of the new blank line where data entry may proceed.

If the pointed line or the line immediately below it is empty no insert will occur, and the null line will be used as the inserted line where data entry may proceed.

To make complex changes to a line already on the screen, the operator may INSERT a line immediately below the original and then retype the line with changes. The original line may then be DELETED.

The pseudo-ENTER key may be used to generate additional blank lines at the same point on the screen.

:L - LOCATE next - typed exactly :L<ENTER>, clears the screen and finds the next line of text. If positioned at the end of the file, the 'next' line will be the first line of the file.

:L <old text> - LOCATE match - similar to FIND match except that the locate command searches for imbedded text matching <old text>. Leading spaces should be supplied if meaningful.

For additional forms of the FIND and LOCATE commands see the 'FILE SEARCH' section.

:M <old text><command separator><new text> - MODIFY - a modify command allows the operator to replace <old text> by <new text>, insert <new text> after <old text> or append (i.e., truncate and add) <new text> after <old text>. For the various forms of this command see the MODIFICATION' section.

:SC - SCRATCH above - in all but Comment mode this command erases the lines from the top of the screen down to the pointed line, inclusive. (In Comment mode, only the comment fields are erased.)

The cursor is left on the pointed line where data entry may proceed.

:SB - SCRATCH below - in all but Comment mode this command erases the lines from the pointed line to the bottom of the screen, inclusive. (In Comment mode, only the comment fields are erased.)

The cursor is left on the pointed line, where data entry may proceed.

:E - END - the end command causes the remainder of the logical source file to be copied to the logical scratch file and then, if the logical scratch is not the physical input file, the scratch file is copied back to the source file.

The command line will be left on the screen as long as the copy from source to scratch is in progress; it is erased during the final copy from scratch back to source.

The end may be terminated as long as the command line is still displayed, by pressing the KEYBOARD and DISPLAY keys. When the final copy is completed, control is returned to the DOS.

Note that if the one-pass option was selected in the parameter list, no copy from scratch back to source will be performed.

:E/ - END/DEL - this command causes the remainder of the source file to be deleted (the lines currently on the screen will be written out), and, if the logical scratch file is not the physical source file, the scratch file is copied back to the source file. When the file is completely updated, the system is reloaded.

No copy back is done if the one-pass option is set.

4. MODIFICATION COMMANDS

4.1 DELETE COMMAND

Modification of a line may be achieved in a variety of ways. The DELETE command enables the user to remove leading information while the MODIFY command may be used to replace imbedded information, insert text into a line or field, or truncate and add new text at a specified point or in a specified field.

:D <old text> - DELETE through - this command deletes all characters from the left edge of the pointed line through the specified <old text>. The remaining characters will be left justified and re-displayed. The cursor returns automatically to the command line.

4.2 MODIFY COMMAND

The general form of the MODIFY command is:

:M[#] [old text]<sep>[new text]

where [#] is a number which extends the meaning of the command and <sep> is the command separator which defines the action of the command. Both [old text] and [new text] fields are optional. If [old text] is omitted, the command will take effect at the left most edge of the pointed line (or at the left edge of the specified field). If the [new text] field is omitted, a null field will be used to execute the modification.

4.2.1 LINE MODIFICATION

The following descriptions are of the line modification version of the MODIFY command

:M [old text] < [new text] - MODIFY (replace) - replace the specified [old text] by the specified [new text]. The less than character (<) is a command separator which indicates replacement and, therefore, the [old text] may not contain this character. If [new text] field is omitted, the old text will simply be deleted and the line will be compressed to the left.

For example to modify the text line:

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'S BACK.

The command: :M BROWN<RED would cause the line to be redisplayed like this:

THE QUICK RED FOX JUMPED OVER THE LAZY DOG'S BACK.

The command: :M .< 1234 TIMES. to the original line would generate a line like:

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG'S BACK 1234 TIMES.

If the replacement causes the line to become longer than 79 characters, the trailing word, in text mode only, will be wrapped around and a new line will be inserted containing the entire last word. If the [new text] is shorter than the [old text] it replaces, the line will be shortened.

After the pointed line is redisplayed, the cursor is returned to the command line.

:M [old text] > [new text] - MODIFY (insert) - the command separator greater than (>) causes the [new text] to be inserted in the pointed line immediately after the [old text].

If the line becomes longer than 79 character, and word wrap around is not in effect, the trailing characters are truncated. If, however, word wrap around is on, the trailing character and last word are inserted on a new line.

:M [old text] \ [new text] or

:M [old text] | [new text] - MODIFY (append) - the vertical bar (|) or backslash (\) command separators cause everything in the pointed line, past the [old text], to be replaced by the [new text].

As in all MODIFY commands, if the pointed line becomes longer than 79 characters, truncation occurs if word wrap around is not implemented.

:M[#] - MODIFY repeat - typed exactly :M<ENTER>, uses the <old text> <sep> <new text> from the last MODIFY command. This is useful when making the same change repeatedly. Note that the field number is not saved, and must, therefore, be supplied if necessary.

:M* - MODIFY display - display the expression entered for the last MODIFY. After the saved command is displayed, the cursor is turned off and the operator must press ENTER to proceed. No MODIFY is actually performed.

4.2.2 FIELD MODIFICATION

In field modification mode, the MODIFY command acts only on a specific field and does not expand or contract the entire line but maintains the integrity of all fields before and after the affected field.

:M<#> [old text]<sep>[new text] - MODIFY field - where the pound sign <#> is a number from 1 to 6 designating the field to be modified (or the starting point to search for matching [old text]). Field 1 is always the left hand margin, thus, in Assembler mode, field 1 is the label, field 2 is the op code, field 3 is the expression and field 4 is the comment. (#) This command may be executed in any of the previous Modify forms. However, modification is performed within the specified field only. As long as the text being modified is unique, field 1 may be specified, since the field number indicates only where to start looking for matching text. (Note that if the field number is omitted, line modification is assumed.

Thus, a replacement or append shorter than the original field will be blank filled and subsequent fields will maintain their position and content. An insertion longer than the specified field will be truncated (with the exception of the last field whenever word wrap around is in effect).

For example, in Assembler mode, the line:

```
LABEL      OP      EXP      COMMENT
```

the label may be deleted by the command:

```
:M1 LABEL\
```

with the resultant line:

```
          OP      EXP      COMMENT
```

Or, the expression field (EXP) could be changed to EXP+1 without disturbing the comment field position, by the command:

```
:M3 EXP>+1
```

which generates:

```
LABEL      OP      EXP+1    COMMENT
```

Note that in the above example, :M1 EXP would also have worked, since EXP occurs only in the third tab field.

To add a comment to a line previously containing none or to replace an existing comment field, enter:

```
:M4 \

```

NOTE Remember when using the repetitious form of the MODIFY command that the field number must be supplied.

5. FILE SEARCH COMMANDS

The FIND and LOCATE commands have several forms and have been separated from the basic command set to better describe them.

Manual, operator controlled, searches may be performed by depressing the KEYBOARD and DISPLAY keys simultaneously to cause data to be fetched from the file and displayed (as long as the keys are pressed) on the screen. To fetch a single line use the Pseudo-ENTER key (DEL shifted). The :EO command performs the same function automatically, i.e., it causes lines to be fetched and displayed until the end of file is reached. To terminate a :EO command, press the KEYBOARD and DISPLAY keys.

To find the end of a file, without displaying the entire file (since the display is time consuming) use the :E* command. This will search for the end of file and display the last eleven lines of data.

:F <old text> - FIND match - the screen is cleared and the input file is searched for a line starting with the specified <old text>. Leading spaces in the input lines will be ignored and should not be entered as part of <old text> (note that this command should be typed exactly :F<SPACE><old text>).

A FIND will wrap entirely around the file (or up to the end of file if the one-pass option is set). If the requested text is not found, the last line on the screen when the FIND was executed will be displayed. A FIND may be stopped by pressing the KEYBOARD and DISPLAY keys.

The <old text> specified for a FIND (or locate) command is saved. The saved match may be redisplayed or used again.

:F<SPACE> - FIND same match - if the FIND command is followed by exactly one space and the ENTER key, the previous FIND (or LOCATE) <old text> will be used for this FIND. In this manner, several occurrences of the same text may be searched for.

:F* - FIND display - the asterisk (*) after the FIND command causes the <old text> of the previous FIND or LOCATE command to be displayed. The cursor is turned off and the operator must press ENTER to proceed. No FIND is performed.

:L - LOCATE next - typed exactly :L<ENTER>, clears the screen and finds the next line of text. If positioned the end of the file, the 'next' line will be the first line of the file.

:L <old text> - LOCATE match - similar to FIND match except that the

locate command searches for imbedded text matching <old text>. Leading spaces should be supplied if meaningful.

:L<space> - LOCATE same match - typed exactly :L<SPACE><ENTER>, uses the <old text> specified by either the previous LOCATE or FIND command to perform a search.

:L* - LOCATE display - display the <old text> entered for the previous LOCATE or FIND command. As in the FIND display, the cursor is turned off and the operator must press ENTER to continue. No LOCATE is actually performed.

6. OTHER COMMANDS

:B - BYPASS - (DOS only) - fetch a line from the file, bypassing end of file or record format error (which would normally be treated as an end of file). Subsequent lines (if not also record format errors) may then be fetched by the normal mechanisms. This command is intended as a recovery tool for use only if the file has been accidentally shortened or contains badly formatted records.

:C - COPY - copies the pointed line to the bottom of the screen, deletes the pointed line and rolls the screen up one line. This command cannot be executed on the top screen line.

The cursor is left on the now null pointed line. Text may be entered at this point (the Pseudo-ENTER and word wrap around, if in effect, will apply). When the ENTER key is finally pressed, the pointer is automatically moved to the following screen line so that a group of lines may be easily copied to another part of the screen.

:T - TAB set - this command enables the user to reset the tab stops during execution. (Not available in Comment mode.) The command causes a line of numbers to be displayed across the bottom of the screen.

The operator should space over to each position where a tabstop is desired and type any non-blank character. A maximum of five tab stops will be set. These tab stops are meaningful during data entry and field modification (:M#) since data within a field may be modified without disturbing the rest of the line.

:X - TEXT - (DOS only) this command implements word wrap around and disables shift key inversion and space insertion after leading periods. It automatically enters the tab set command (:T), so that tab stops may be cleared by the operator. The tab key character is not changed; therefore, the :<tab key> command must be used to set a new tab key character if one is desired.

:<tab key> - change tab key character to any non-alpha, non-numeric, non-COLON, non-ENTER character typed after a leading colon on the command line.

7. RECOVER PROCEDURES

7.1 CASSETTE RECOVERY

Parity errors reflect a hardware or physical tape problem. When a parity error occurs, the message 'PARITY ERROR' will appear on the bottom line of the screen and a beep will sound. Processing will not resume until the KEYBOARD and DISPLAY keys are pressed simultaneously.

7.1.1 Parity Errors During Editing

If the parity error occurs during a manual fetch or during a :F, :L or :EO command, press the KEYBOARD and DISPLAY keys to recover the data. Look at the erroneous data (since characters are blocked on the tape, the error could be anywhere in the next 250 characters) and correct. Then proceed with normal data entry.

7.1.2 Parity Error During Copy Back

If the parity error occurs during the final pass of an END or END/DEL command or while performing a DUP, the record containing the error will be copied to the source tape (after the operator presses the KEYBOARD and DISPLAY keys) and further editing will have to be done to correct the error.

7.1.3 Power Failure or Abort

Recovery procedures using the DUP tape designation can be implemented should the edit program be aborted during its execution without having completed all necessary copying and end-of-file writing.

Causes of difficulties during execution of the program which could require such action are:

1. Power failure.
2. Turning off the power.
3. Pressing the RESTART key.
4. Encountering unrecoverable tape errors.
5. Removal of the tape cassette.

7.2 DOS RECOVERY

A 'FORMAT TRAP' occurs when a record not belonging to the current file is encountered. This can be caused either by a physical misalignment of the disk read head or because a

record has erroneously been written into that file by some other program.

A 'RANGE TRAP' occurs when the physical limit of the file is reached and no end of file is present.

7.2.1 Bypassing Errors or End of File

When a format or range error occurs, an appropriate message appears on the command line and the cursor is turned off. In order to proceed, the operator must first press the DISPLAY key. The effect of either a format or range trap is the same as an end of file and no further data will be read from the file.

To read past a format error or past an end of file, use the BYPASS command, :B, repeatedly if necessary.

7.2.2 File Recovery

If the source file is lost (e.g., erroneously KILLED), the scratch file may contain a useful copy. The scratch file (SCRATCH/TXT) contains a copy of the last file edited, it may be used to recover only that file.

8. GLOSSARY

- Assembler mode - assumed mode of execution. Tab stops at 9, 15 and 30 (may be changed during execution). The space bar is assumed as the tab key character (this may be changed in parameter list or during execution). Shift key inversion and no word wrap around are assumed. Leading period (.) generates period space (.) for comment lines. Pseudo-ENTER does line-insert.
- Command - characters typed at the left edge of the command line following a COLON (:) which have special meaning to the editor.
- Command line - the twelfth line of the screen where most data is entered, lines are fetched and commands are typed.
- Command separator - the character in a MODIFY command which indicates what is to be done (> means insert, < means replace and \ or | mean append).
- Comment field - in assembler code the area of the screen from columns 30 to 79 is generally used for programmer comments.
- Comment mode - executed if 'C' in parameter list. Facilitates changing or adding comments to assembler code. Tab stops at 9, 15 and 30 (may not be changed during execution). The space bar is assumed to be the tab key character (this may be changed in parameter list or during execution). Shift key inversion and no word wrap around are assumed. Leading period (.) generates period space (.) for comment lines. Pseudo-ENTER positions to comment field of following line and deletes the comment. Delete and Scratch commands affect only the comment field. Trailing blanks are truncated when data is output.
- DATABUS mode - executed if 'D' in parameter list. Tab stops at 9 and 15 (may be changed during execution). The space bar is assumed to be the tab key character (this may be changed in the parameter list or during execution). Shift key inversion and no word wrap around are assumed. Leading period (.) generates period space (.) for comment lines. Pseudo-ENTER does line-insert. Input lines are blank filled and trailing blanks are truncated on output.
- Field number - a digit used in the MODIFY command to designate characters between two tab stops. Field '1' is always from column 1 to the first tabstop; thus, in Assembler mode, '1' designates the label field, '2' the opcode field, '3' the expression field and '4' the comment field. During field modification, trailing fields are preserved.

Format trap - bad record encountered on disk. See 'RECOVERY PROCEDURES', section 7.

Line insert - results from an INSERT command, data entry or modification when word wrap around is in effect or a Pseudo-ENTER key in any mode other than Comment. The lines above the pointed line are rolled up and a new, blank line is generated at the pointed line.

Logical scratch file - current output file.

Logical source file - current input file.

New text - a group of characters, typed immediately after a command separator in a modify command, which will become part of the line being modified.

Old text - a group of characters, including spaces, which are searched for, either in the pointed line (as in the MODIFY command) or in the file (as in the FIND or LOCATE commands).

One-pass option - a flag may be set in the parameter list which inhibits updating of the original source file. The FIND, LOCATE and END, END/DEL commands will not write back into the input file if this option is set.

Parameter list - initialization information provided when the editor is first executed. Following file specifications, a SEMI-COLON (;) indicates the presence of a parameter list. The mode, one-pass option, tab character, margin bell column and (in text mode) 'no shift inversion' (S) and 'no word wrap around' (L) may be set.

Pointed line - a pointer (>) in the left hand margin is used to reference lines for modification by command. The line to the right of the pointer is the pointed line.

Physical scratch file - specified (or implied SCRATCH/TXT) output file (or rear cassette).

Physical source file - specified input file (or front cassette).

Pseudo-ENTER - the key marked DEL (always shifted) is referred to as the Pseudo-ENTER key. If pressed in the first column of the command line, one line of text will be fetched from the source file.

In comment mode, if pressed on any but the bottom screen line or command line, it will cause the cursor to be positioned to the comment field of the following line and that field will be erased.

In all other modes, the Pseudo-ENTER key causes a new line to

be inserted so that data entry may proceed in the same area of the screen. If pressed on the last screen line, the Pseudo-ENTER key simply places the cursor on the command line.

Range trap - attempt to read or write past allocated space on disk - see 'RECOVER PROCEDURES' in Section 7.

Scratch file - at any point in time, the logical scratch file is the output file. It may, however, physically be the original input or the assigned 'scratch' file.

Screen line - any of the eleven lines on the screen which may be referenced by the command pointer. The command line is not, therefore, included.

Shift key inversion - reverse the function of the shift key for all alpha characters so that, in lower case, alpha characters will appear upper case.

Source file - originally this is the input file specified at initial execution. The term source file refers to the current input file; thus, at any point in time, the logical source file may be either the specified input file or the file specified as the scratch file.

Text mode - executed by a 'T' in the parameter list. No tab stops are set (tabs may be set during execution). The SEMI-COLON (;) is the assumed tab character (the tab key character may be changed in the parameter list or during execution). No shift key inversion is performed (this may be selected in the parameter list). Word wrap around is performed (this feature may be turned off by an 'L' in the parameter list).

Word - a word is defined as any group of less than 50 characters preceded by a space.

Word wrap around - a feature of text mode. During data entry a space within the last 10 columns of the screen causes an immediate carriage return. If this occurs on a screen line, a line insert is performed so that data entry may proceed at the same area of the screen. If a character is typed over the last column of the screen, the last word is removed, a line insert performed and the removed word is placed at the beginning of the inserted line where data entry may proceed. If a modify command causes the line to become longer than 79 characters, the trailing characters, including the last word on the line, will be moved to a new line which will be inserted below the original line. Control will then return to the command line.

Write edit format - A DATABUS compatible cassette format selected in the parameter list. This tape format creates one

80-character tape record per screen line. Spaces are not compressed and trailing spaces are supplied to fill the line out to 80 characters.

9. COMMAND LIST

:B BYPASS end of file

:C COPY pointed line to command line and roll up

:D DELETE entire line

:D <old text> DELETE from left thru <old text>

:E END edit - copy remainder of file and update source

:EO EOF display - fetch and display data until end of file

:E/ END/DELETE update without copying remainder

:E* EOF search - find end of file and display last full screen

:F <old text> FIND match - search file for matching leading text

:F<SPACE> FIND repeat - use previous find/locate <old text>

:F* FIND display - display previous find/locate <old text>

:I INSERT a blank line below pointed line

:L LOCATE next - clear screen and get next line

:L <old text> LOCATE match - search file for matching imbedded text

:L<SPACE> LOCATE repeat - user previous find/locate <old text>

:L* LOCATE display - display previous find/locate <old text>

LINE MODIFICATION

:M [old text]<[new text] - MODIFY replaces old text by new text, adjusting the entire line

:M [old text]>[new text] - MODIFY inserts new text after old text, adjusting the entire line

:M [old text]\[new text] or :M [old text]||[new text] - MODIFY appends new text after old text adjusting the entire line

FIELD MODIFICATION

- :M<#> [old text]<[new text] - field MODIFY replaces old text within specified field with new text without disturbing the remainder of the line.
- :M<#> [old text]>[new text] - field MODIFY inserts old text after new text within specified field, without disturbing the remainder of the line.
- :M<#> [old text]\[new text] or :M<#> [old text]|[new text] - field MODIFY appends the new text after the old text within the specified field without disturbing the remainder of the line.

- :M* MODIFY displays the previous modify [old]<sep>[new]
- :M[#] MODIFY repeats the previous modify [old]<sep>[new]
- :SB SCRATCH BELOW deletes the pointed line and all screen lines below it
- :SC SCRATCH ABOVE deletes the pointed line and all screen lines above it
- :T TAB SET permits the user to set up to five tab stops
- :X TEXT mode (DOS only) switches to text mode with word wrap around and no shift key inversion.
- :<character> changes the tab key character to <character>