August 1978

This document contains information on the two text-editing programs provided with VAX/VMS. It describes in detail the features and use of SOS, an interactive editor, and SLP, a batch-oriented editor.

VAX-11
Text Editing
Reference Manual

Order No. AA-D029A-TE

SUPERSESSION/UPDATE INFORMATION:	This is a new document for this release.
OPERATING SYSTEM AND VERSION:	VAX/VMS V01
SOFTWARE VERSION:	VAX/VMS V01

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation • maynard, massachusetts

First Printing, August 1978

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI

CONTENTS

CONTENTS (Cont.)

FIGURES

TABLES

Page

TABLES (Cont.)

PREFACE

MANUAL OBJECTIVES

This manual is designed as a reference for users who have had some exposure to text-editing computer programs. It is not intended as a tutorial introduction, either to text editors in general or to the SOS and SLP editors in particular.

INTENDED AUDIENCE

If you have read the introduction to SOS in the VAX/VMS Primer and want to know more about SOS, this is the manual for you.

If you are familiar with some other text editing program -- either a DIGITAL text editor or another -- you can use this manual to gain familiarity with SOS and SLP. The tables and figures will prove helpful in this familiarization. Later, you can study this manual to learn details of SOS and SLP commands and features.

STRUCTURE OF THIS MANUAL

Information in this document is organized as follows.

- Chapter 1 is a brief introduction to text editing; it touches on some of the differences between SOS and SLP.
- Chapter 2 describes SOS in detail.
- Chapter 3 describes SLP in detail.
- Appendix A summarizes SOS functions and features, and Appendix B lists SOS error messages.
- Appendix C summarizes SLP commands, and Appendix D lists SLP error messages.

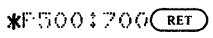
ASSOCIATED DOCUMENTS

The following documents may also be useful.

- VAX-11 Information Directory
- VAX/VMS Primer

CONVENTIONS USED IN THIS MANUAL

Colored ink is used in examples of user-system dialog to show the text that you type. For example:

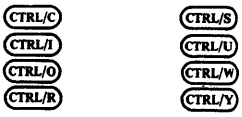


This means that after the SOS prompt (*) you type P500:700 followed by a carriage return.

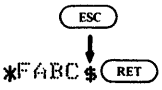
The following symbols represent special keys on the terminal keyboard:

- (ESC) The escape key (labeled ESC, ALT, SEL, or PRE, depending on the terminal)
- (RET) The carriage return key (labeled CR or RETURN)
- (DEL) The delete key (labeled DEL, RUB, or CAN)
- (LF) The line feed key
- (SP) The space bar
- (TAB) The tab key or CTRL/I (see below)
- (BSP) The backspace key

The following symbols stand for various control functions, which you initiate by pressing the given key while holding down the CTRL key. Section 2.1.2 discusses these control functions.



A downward-pointing arrow (↓) marks a character that is not echoed literally. For example, SOS echoes the escape character as a dollar sign ($). In Chapter 2, this is indicated as follows:



CHAPTER 1

INTRODUCTION TO TEXT EDITING

Text editing on a computer system means creating, modifying, and maintaining files of information. The files can contain any kind of character-oriented information. Some examples are:

- Source language for an assembly language or higher-level language program
- Clear text -- such as memos, correspondence, documentation
- Data -- such as records from a payroll master file or columns of numbers for input to a chemical-analysis program

Text editing does not refer to maintaining binary files, image files, or any other non-character-oriented information.

1.1 SOS AND SLP

This manual describes two text editors, which are computer programs you can use to create, modify, and maintain ASCII files:

- SOS is an interactive editor. Using it, you work online to the computer, carrying on a dialog with the program as you create or modify a file. (SOS stands for Son of STOPGAP, an earlier text editor.)
- SLP is a batch-oriented editor. It is designed to make changes to a text file in one pass, without your intervention, guided by a file of instructions you have prepared. (SLP originally meant "source language input program.")

Interactive editing, SOS-style, is the usual way to create files. SOS is also suitable for modifying a file interactively -- for thinking as you work. However, this method of text editing, while more amenable to creative thought, does not leave a record of how a file was modified.

If you need such a record, a batch-oriented editor is more suitable. Use SLP if you need an exact record of what changes were made to a file, and when. SLP is widely used in PDP-11 program-development environments. Changes to a given reference copy (a "base level") of an operating system, for example, can be succinctly described and distributed by means of an SLP command file applied to the base-level source files.

A batch-oriented editor is also useful when you want to check the editing commands carefully before issuing them, or when you want to minimize the editing time on a system.

SOS is often used to create SLP command files. Many people use SOS for almost all their text-maintenance tasks, relying on SLP only when they need a formal and complete record of changes to a file.

SOS and SLP are both invoked by the DCL command EDIT. The default editor is SOS. Therefore, to start SOS, type EDIT or EDIT/SOS; to start SLP, type EDIT/SLP.

1.2 LINE NUMBERS

Different text editing programs use different methods to refer to the text they work on. Some use line numbers, whereas others refer to text lines strictly by their content. For example, a line-oriented text editor might accept the command "delete line 50," while a content-oriented editor would take the command "find the next line that contains ABC and delete that line."

SOS and SLP both refer to text by line number.[1] But they differ in the philosophy behind their line numbering; as a user, you must be aware of the differences.

In SOS, line numbers are actually part of each text line. When SOS writes your file to disk, it includes the line numbers unless you specifically request that it omit them. Lines are numbered with some interval between them (usually 100), and new lines that you insert must fit between adjacent line numbers. Unless you give a command to renumber part of a file, any given line of text has the same line number each time you edit the file with SOS.

In SLP, by contrast, line numbers are not part of the text. The number of a line in a file you are editing with SLP corresponds to its sequential position relative to the beginning of the file. Thus, a given text line may have different sequential numbers for different runs of SLP, if lines have been inserted or deleted at an earlier point in the file.

In Chapters 2 and 3, this distinction is made by referring to "line numbers" in the SOS discussion (Chapter 2), and to "sequence numbers" in the SLP discussion (Chapter 3).

---

1. Both editors also have features that let you search for a given string in the text.

CHAPTER 2

INTERACTIVE TEXT EDITOR (SOS)

2.1 INTRODUCTION TO SOS

SOS is a line-oriented, interactive, text-editing program. Unlike many line-oriented editors, SOS has features that allow you to examine and change text character by character. You can use SOS to perform the following functions:

- Examine, create, and modify ASCII text files
- Search for and/or change one or more arbitrary text strings, with the option to verify each change before it is made
- Merge parts of one file into another
- Create a file that is a subset of another file

SOS is line-oriented, so it works with line-numbered text files. If you edit a file that does not have line numbers, the editor adds line numbers to the text lines.

Unlike some (content-oriented) text editors, SOS requires you to be concerned with line numbers within the file. For most SOS commands, a line number or a range of line numbers specifies the text to be operated on. When you insert, delete, move, or copy text, SOS is programmed to keep line numbers in ascending order within each "page" of text. To this end, SOS may either refuse to accept more inserted text, insert "page marks," or warn you to renumber the lines in the file.

SOS is more interactive than many other editors. In some modes of operation it responds on a character-by-character basis to what you type. This degree of interactivity may be new to some users accustomed to less interactive editors, which always expect a command line terminated by a carriage return.

For example, one SOS feature that exhibits this character-by-character interactivity is Alter mode. This special mode lets you make changes within a line of text, interactively. Alter mode has its own commands and syntax; it functions essentially as an editor within an editor. Alter mode is discussed in detail in Section 2.8.1.

Advanced features of SOS allow you considerable flexibility in searching for a string of text (see Section 2.10) and allow you to specify blocks of text by content, instead of by line number (see Section 2.11).

Many SOS parameters and switches, which you can set, control the editor's default values (see Sections 2.8.21 and 2.9).

2.1.1 Initiating and Terminating SOS

Initiate SOS by typing one of the following commands in response to the DCL prompt:

```
$ EDIT(RET)
```

or

```
$ EDIT file-spec(RET)
```

or

```
$ EDIT/OUTPUT:output-file-spec file-spec(RET)
```

In the first form above, SOS prompts you for the specification of the file you want to edit or create. In the second form, you explicitly type the file specification (file-spec). If the specified file does not exist, SOS creates it. In the third form, outfile is the specification for an output file.

In the second and third forms above, you can optionally append qualifiers, separated by slashes (/), to the file specification.

SOS does not supply a default file type. If you omit it, the file type is null.

There are several ways to terminate SOS; they are discussed in detail under the End command (E) in Section 2.8.4. Usually, you give the command E after SOS's prompt (*):

```
*E(RET)
[file-spec]

$
```

Upon terminating, SOS writes an output file containing all the modifications you made in editing the file. The original file is not changed. The specification SOS uses for the output file (unless you supply it explicitly) has a version number higher by 1 than the latest version of the original file.

2.1.2 Control Characters

All the usual VAX/VMS control characters can be used in most contexts within SOS. However, some of these characters are not effective in Alter mode. For details, see the Alter command in Section 2.8.1. The control characters are described briefly below.

(DEL) (DELete, or RUBout, or CANcel) This character deletes the previous input character. Successive (DEL) characters remove previous characters one by one. Any number of characters can be deleted, up to the last carriage return or escape typed. The way this deletion is represented on your terminal depends on the type of terminal. On CRT terminals, the characters are physically deleted and the cursor is moved backward. On hard-copy terminals, the deleted characters are echoed in reverse order, and set off within backslashes (\).

(CTRL/U) This character, typed before an input line is complete, deletes the entire line being input. You can then retype the line. (CTRL/U) is generated by typing u or U while holding down the CTRL key. It is echoed as ^U.

(CTRL/S) This character, typed while output is in progress, suspends the output until you type (CTRL/Q). (CTRL/S) is not echoed.

CTRL/Q This character resumes output suspended by CTRL/S. CTRL/Q is not echoed.

CTRL/O This character interrupts output in progress. SOS continues to send output to the terminal, but the terminal discards it. SOS prompts when it is ready for further input; the prompt depends on which mode the editor is in. CTRL/O is not echoed.

CTRL/R This character, typed before an input line is completed, redisplays the entire line. CTRL/R is echoed as ^R.

CTRL/Y This character immediately gets the attention of the DCL interpreter. You can then type any legal DCL command, such as SHOW or CONTINUE. If the DCL command does not change the SOS image, you can resume SOS by typing the DCL command CONTINUE.

CTRL/C This character interrupts whatever command is in progress and returns you immediately to Edit mode. Any changes already made by the command (for example, by an Alter or Substitute command) are incorporated into the file. CTRL/C is echoed as ^C.

2.1.3 Use of Escape

Some SOS commands use the escape character, ESC, as part of their syntax. On some terminals, the key corresponding to the escape character is labeled ESC; on others, it is labeled ALT (altmode), SEL (select), or PRE (prefix).

The escape character is a message delimiter in the VAX/VMS system. Although many VAX-11 programs accept escape interchangeably with carriage return to indicate the end of input, SOS does not. You can use escape only in the contexts in which it is legal.

In this manual, ESC indicates that you should press the escape key on the terminal.

SOS echoes the escape as $. Do not confuse this character with the dollar-sign character, shift-4. The dollar sign normally has no special meaning to SOS.[1]

2.1.4 Upper- and Lowercase Characters

Whether a terminal generates all uppercase characters, or both upper- and lowercase characters, depends on the type of terminal and on how the terminal characteristics are set. (Terminal characteristics are changed by the DCL SET command -- see the VAX/VMS Command Language User's Guide and the VAX/VMS Primer.) SOS accepts either uppercase or lowercase characters in commands, or a combination of the two. In the examples in this manual, upper- and lowercase characters are used interchangeably in commands. SOS distinguishes between upper- and lowercase characters in text that you enter into a file.

---

1. The dollar-sign character is significant only when the parameter ESCAPE is set, enabling you to insert escape characters into a text file. See Sections 2.9.1 and 2.8.22.

Normally, SOS does not distinguish uppercase from lowercase when searching for a given text string. Thus, a search for the text string "holiday," using the Find or Substitute commands, would be satisfied if SOS encountered any of the following: holiday, Holiday, or HOLIDAY. (See Section 2.8 for a description of these search commands.) You can request SOS to distinguish upper- and lowercase characters either by setting the EXACT switch (see Section 2.9.2) or by using an option of the Find and Substitute commands.

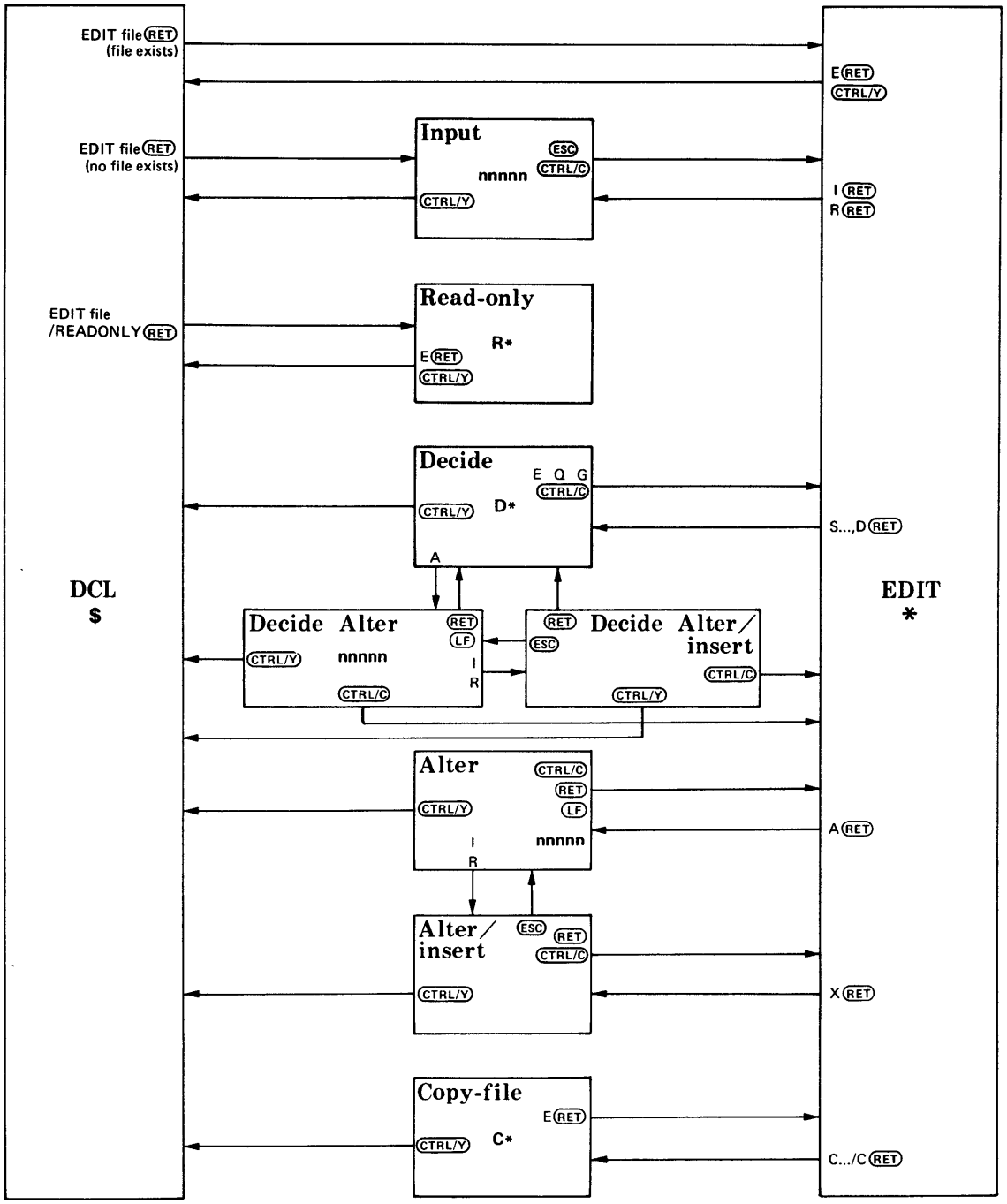
2.2 MODES OF OPERATION

SOS operates in a number of different modes. A **mode of operation** is defined as a state of SOS in which the editor interprets terminal input in a distinctive way. You can tell which mode SOS is operating in by the way the terminal behaves.

Figure 2-1 represents the seven modes and two submodes of SOS operation, the paths among them, and the prompt for each mode. You can initiate the editor in one of three modes: Input, Read-only, or Edit. Edit mode is the heart of SOS, from which the other modes can be accessed. Two submodes of Decide mode correspond to Alter and Alter/insert modes, but communicate only with Decide mode. Throughout the rest of this chapter, miniature versions of Figure 2-1 mark places in the text where the various modes are discussed.

Figure 2-1 also shows the command(s) that let you change from one mode of operation to another. These commands are described in Section 2.8.

In different modes, SOS prints different prompts when it is ready to accept input, as shown in Figure 2-1. It then accepts a certain set of commands, each set unique to its mode. In some modes, SOS interprets input on a character-by-character basis, so that you need not type (RET) after a command. In different modes, SOS has different ways of informing you that you have typed unacceptable input.



The paths among the various SOS modes and submodes of operation are marked by arrows. Prompts are shown in boldface type.

Figure 2-1 SOS Modes of Operation

As mentioned above, you can initiate SOS in one of three modes:

- In Input mode, SOS accepts text you type and adds it to the file.
- In Read-only mode, you can only search and display the file; you cannot change it.
- In Edit mode, you issue most of the commands to display and change a file. From Edit mode you can move to any other mode, or you can exit to DCL.

Four other modes of SOS operation are accessible from Edit mode. They are Copy-file, Decide, Alter, and Alter/insert.

- Copy-file mode lets you to copy all or part of one file into another. You can search and display the copy file to locate the text to be copied.
- Decide mode is used in conjunction with the Substitute command (see Section 2.8.15). You can issue a Substitute command to replace all occurrences of one text string with another string within a given range of text. If you specify Decide mode, you can choose on a case-by-case basis whether or not to make the substitution, or you can enter Alter mode to make more extensive changes within a line.
- Alter mode lets you edit character by character within a line. In this special intraline mode, each character that you type is a command that SOS acts upon immediately; you need not type (RET).
- Alter/insert mode lets you insert text that you type into the text file. In this mode, most control characters are treated as text; they do not perform their usual control functions.

Decide mode has two submodes: Decide Alter and Decide Alter/insert. In these submodes, SOS acts almost exactly as it does in Alter mode and Alter/insert mode, respectively. The only difference is that these submodes are accessed from Decide mode, not from Edit mode. When you exit from the submodes you return to Decide mode, not to Edit mode.

The following sections discuss the seven modes and two submodes of SOS in general terms. More detail on each mode of operation is given later, at places in the text marked by miniature copies of Figure 2-1.

2.2.1 Input Mode

In SOS Input mode you can insert one or more new lines of text into a file. You enter Input mode either directly from DCL or from Edit mode.

SOS starts up in Input mode if, when you give the DCL EDIT command, you specify a file that does not exist. SOS creates a file with the specified name and waits for you to type text to put into it. An example follows.

```
$ EDIT NEW.FOR(RET)

INPUT:DBA1:[MANCH]NEW.FOR;1

00100
```

Notice that SOS types the word INPUT before the filespec to remind you that it is creating the file and entering Input mode.

In Input mode, the prompt is the line number that SOS assigns to the next line to be typed. Initially, SOS numbers lines by 100s.

To correct typing errors while entering text, use any of the terminal control characters described in Section 2.1.2.

When input is complete, switch to Edit mode by typing an escape character. You can type escape either at the end of a line of input, or after SOS has prompted with the next line number. SOS then prompts with an asterisk (*), indicating Edit mode.

You can store the file and end SOS at this point by typing the command E. Or, before terminating, you can make changes to the text you have just typed by using other Edit-mode and Alter-mode commands.

To enter Input mode again -- if you have more whole lines of text to put into the file -- use the Input or Replace commands (described in Sections 2.8.7 and 2.8.14, respectively).

2.2.2 Read-only Mode

Use Read-only mode if you only need to examine a file and do not intend to modify it, or if you need to look at a file that is protected against your writing it. Start SOS in this mode by appending the qualifier /READONLY to the filespec:

```
$ EDIT PAYROLL.MST/READONLY(RET)

READ:DBA1:[CORP]PAYROLL.MST;3
R*
```

Note that SOS types the word READ before the filespec, and that R* is the prompt in Read-only mode.

Only nine commands are legal in Read-only mode (see Section 2.8): End, Find, List, Print, Save World, Move Position, Give Parameter, Set Parameter, and Command File. These commands do not modify the file. Any other commands, which would modify the file, result in the error message COMMAND INVALID FOR READ ONLY FILE.

Read-only mode is the most efficient way to use SOS if you do not need to modify a file. It is efficient because SOS does not copy the file to a working file as it does in Edit mode. See Section 2.13 for details.

2.2.3 Edit Mode

Edit mode constitutes the main part of SOS. You can switch to any of five other modes of operation from Edit mode, and you can return to Edit mode from any of them. (You cannot switch to Read-only mode from Edit mode, or vice versa.)

SOS accepts 24 commands in Edit mode. Most are represented by a single character, and most take arguments in a predetermined syntax. Table 2-1 lists the names and single-character forms of the commands, and gives a brief summary of their functions. For complete descriptions of the commands, see Section 2.8.

Initiate SOS in Edit mode by giving the filespec of a file that already exists, either on the EDIT command line or in response to SOS's prompt. For example:

```
$ EDIT (RET)
FILE:TESTBLD.MAC (RET)

EDIT:DBA1:[HENDRIKS]TESTBLD.MAC;6
*
```

SOS prints the word EDIT before the filespec, indicating Edit mode. The prompt for this mode is an asterisk (*).

For details on the operation of each Edit-mode command, see Section 2.8. Table 2-3 in that section summarizes the command syntax. Most commands take as an argument a line number, or a range of line numbers, indicating what text SOS is to operate on. Page numbers may also appear if the file is divided into pages. The formats of line numbers, page numbers, and ranges are given in Section 2.3.

Table 2-1
Edit-Mode Commands: Form and Function

Form	Command	Meaning
A	Alter	Enter Alter mode for intraline, character-by-character editing.
C	Copy	Copy a range of lines to another place within a file, or from another file. Optionally, enter Copy-file mode.
D	Delete	Delete a range of lines.
E	End	End SOS, return to DCL.
F	Find	Search for the occurrence of one or more specified strings of text.
H	Help	Print Help package on terminal.
I	Input	Enter Input mode to insert lines of text.
J	Join	Join two text lines into one line.
K	Kill page mark	Delete a page mark.
L	List	List a range of lines on the printer or to a file.
M	Mark	Insert a page mark.
N	reNumber	Renumber a range of lines.
P	Print	Print a range of lines on the terminal.
R	Replace	Delete a range of lines and enter Input mode.
S	Substitute	Replace one or more text strings with other string(s) in a range of lines. Optionally, enter Decide mode.
T	Transfer	Copy a range of lines to a new location and delete the original lines.
W	Save World	Write a new file containing all the changes made so far.
X	eXtend	Enter Alter/insert mode to add text to the end of a line.
.	Move Position	Reset the position of the current line.
=	Give Parameter	Give the value of an SOS internal parameter or switch.
/	Set Parameter	Reset an SOS parameter or switch.
@	Command File	Execute the SOS commands contained in a command file.
(RET)		Print next line.
(ESC)		Print previous line.

2.2.4 Copy-file Mode

Copy-file mode allows you to copy all or part of one file into another file at a specified location. Furthermore, it lets you search through and display the copy file to locate the exact part you want to copy.

Enter Copy-file mode by using the Edit-mode command Copy with the qualifier /C (see Section 2.8.2). For example, to search the file SUBRLIB.FOR and copy part of it after line 100 into the file being edited, type:

```
*C100=SUBRLIB.FOR/C (RET)
C*
```

Note that C* is the prompt for this mode. The copy file is opened read-only. Thus, only commands that do not change the file are legal, as in Read-only mode. The commands most often used are Print and Find.

To exit from Copy-file mode, use the command E. (Do not confuse this command with the Edit-mode command E, with which you exit to DCL.) SOS then prompts for the range of lines to copy. When you supply the range, the copy operation is completed and you return to Edit mode.

Copy-file mode is described in detail in Section 2.8.2.1.

2.2.5 Decide Mode

The SOS command Substitute (see Section 2.8.15) allows you to replace arbitrary strings of text with other strings. Decide mode, entered by way of the Substitute command, lets you decide, line by line, whether or not to make the specified substitution.

For each line in which a substitution is called for, SOS displays the modified line and prompts with D* for your decision. You reply with a single character. (No (RET) is needed; SOS reads and acts upon the single character that you type.) If you reply with an illegal character, SOS repeats the prompt.

The options you can exercise in Decide mode include:

- Yes, make the change as shown.
- No, do not make the change.
- Enter Decide Alter mode with the change already made.
- Make the change, and make all further changes automatically, without decision.
- Return immediately to Edit mode.

Decide mode is described in detail in Section 2.8.15.3.

2.2.6 Alter Mode

Use Alter mode to make minor changes within a line of text. In this mode, you can perform the following functions:

- Insert or delete single or multiple characters or words within a line
- Move forward or backward within a line, either by character(s) or word(s)
- Move to the next occurrence of a given character within a line
- Delete all text until the next occurrence of a given character within a line
- Split a line at any point
- Insert control characters into a line

In Alter mode, SOS does not echo single-character commands that you type; if you type an illegal command, a bell rings on the terminal. SOS displays the line of text that you are editing as you build it, that is, as you insert, delete, and pass over characters and words.

You enter Alter mode by issuing the Alter command from Edit mode. The prompt in Alter mode is the line number followed by a tab, the same as the prompt for Input mode. You can easily determine which mode SOS is in, however, by observing the behavior of the terminal (for example, commands are not echoed in Alter mode).

There are several ways to return to Edit mode. The most common is to type (RET). SOS prints the remainder of the line, if any, and gives the Edit-mode prompt (*).

Alter mode is described in detail in Section 2.8.1.

2.2.7 Alter/Insert Mode

In Alter/insert mode, characters that you type are inserted into the text. Most control characters (with seven exceptions[1]) are inserted into the text without performing their usual special functions. For example, (CTRL/U) does not discard the input line. (See the VAX/VMS Command Language User's Guide for control-character functions.)

You enter Alter/insert mode from Alter mode by giving the I or R command, or from Edit mode by giving the eXtend command. SOS does not prompt in Alter/insert mode. You return to Alter mode by typing (ESC), or to Edit mode by typing (RET).

Alter/insert mode is described with the Alter-mode command I in Section 2.8.1.

1. The exceptions are (RET), (ESC), (DEL), (LF), (CTRL/C), (CTRL/R), and (CTRL/Y).

2.3 LINE NUMBERS, PAGE NUMBERS, AND RANGES

Many SOS commands take as one argument a line number or a range of line numbers, indicating what text the command is to operate on. This section describes how to use line numbers and page numbers to form such ranges. It also discusses shorthand characters for line and page numbers, line and page number offsets, and SOS-supplied defaults in range specifications. The examples in this section use the Print command, which displays lines of text.

2.3.1 Line Numbers

SOS line numbers are 5-digit decimal numbers, such as 00400 or 13280. Line numbers may not be larger than 65535. When you type a line number, you need not type leading zeros: thus typing 50 is the same as typing 00050.

When SOS first numbers a file, it assigns line numbers starting at 100 and ascends by hundreds. (You can change these defaults with the Set Parameter command; see Sections 2.8.21 and 2.9.1.) Line numbers are always in ascending order on any given page.

The following is an example of the Print command using a single line number:

```
*P300(RET)
00300    ;    DATE: 15-AUG-77
*
```

2.3.2 Page Numbers

If a file is large, or if the information it contains falls naturally into a number of logical divisions, you can divide the file into pages. For example, if an assembly-language source file contains a number of logically independent sections, you could place each section in this file on a separate page.

The Edit-mode commands Mark and Kill Page Mark, respectively, insert and delete page marks in a file (see Sections 2.8.11 and 2.8.9). Sometimes SOS automatically inserts page marks when you give a Copy or Transfer command, depending on how the line numbers fall.

The complete specification for a line position in an SOS text file can contain both a line number and a page number. A slash (/) separates the two. For example:

```
*P110/2(RET)
```

This means print line 110 on page 2.

2.3.3 Positions

A **position** marks a single line in a text file. Some commands take a position as an argument, as for example the Mark command (Section 2.8.11). Other commands take a range as an argument (see the next section); a range can be specified by giving two positions as its end points.

A position consists of a line number and a page number. You can omit either one; if you do, SOS supplies a default, as shown below. The notation => means "is defined as." The braces indicate that one of the enclosed expressions must be chosen.

```
                 ⎧ line      ⎫   (current page assumed)
position => ⎨     /page  ⎬   (first line on page assumed)
                 ⎩ line/page ⎭
```

2.3.4 Ranges of Line Numbers

A range of line numbers can be formed in two ways.[1] One way is to specify a starting position and an ending position; separate the two positions with a colon (:). For example, the following command tells SOS to print all lines from line 100 on page 4 through line 500 on page 5, inclusive.

```
*P100/4:500/5(RET)
```

Depending on how the lines are numbered, this command could tell SOS to print five lines, or 50, or none. If you know exactly how many lines you want to print, but not what all their numbers are, you can form a line-number range using the exclamation point (!) to indicate a relative number of lines. The following command tells SOS to print six consecutive lines, starting with line number 100 on page 4.

```
*P100/4!6(RET)
```

This command prints six lines whether or not it encounters a page boundary.

The end points of a range need not themselves be existing line numbers. For instance, in a file whose lines are numbered by 10s, the range 33:67 refers to the three lines numbered 40, 50, and 60.

A range can consist of only a single line: you can omit the ending position or relative number of lines. Previous examples showed how the Print command, which takes a range as its argument, can print a single line.

A command whose range is /page applies to all lines on the page.

You can omit the starting position of a range; SOS defaults the starting position to the current line, as shown below.

```
                 ⎧ position                ⎫   (range is a single line)
                 ⎪ position : position     ⎪
range => ⎨ position ! n              ⎬
                 ⎪          : position     ⎪   (range starts at current line)
                 ⎩          ! n           ⎭   (range starts at current line)
```

---

1. A third, much less common way to form ranges is described in Section 2.11.

2.3.5 Shorthand Characters

SOS recognizes three characters as shorthand for special line numbers. These shorthand characters can be used in position specifications to stand for both line numbers and page numbers:

. (period)	The current line
/.	The current page
^ (caret or up-arrow)	The first line on a page
/^	The first page in the file
* (asterisk)	The last line on a page
/*	The last page in the file

The character ^, ASCII 94, is represented as a caret on some terminals and as an up-arrow on others.

The character 0 can be used interchangeably with ^ to represent the first line on a page or the first page in a file. A line or page number of 0 is illegal, and SOS interprets 0 in a position specification to mean ^.

Some examples of the use of these shorthand characters follow.

*P100:.RET	Print all lines from line 100 through the current line, inclusive.
*P^:620RET	Print all lines from the beginning of the page through line 620.
*P .:.10RET	Print 10 lines beginning with the current line.
*P^:*RET	Print all lines on the current page.
*P/.RET	Same as above: print all lines on the current page.
P:2RET	Print the last line on the current page and the first line on the next page.

2.3.6 The Current Position

The **current position** in SOS is the position of the last line referred to by a command.[1]

For example, after SOS completes a Print or List command, the current position is that of the last line that was output. After a Copy or Transfer, it is the last line moved. After a Find or Substitute, it is the last line matched. (These commands are described in Section 2.8.) Immediately after SOS is initiated, the current position corresponds to the first line in the file.

---

1. The terms "current position" and "current line" are used interchangeably. In fact, the current position is made up of two components, the current line number and the current page number.

The current-line pointer is not moved if a command fails for any reason and returns an error message.

Two Edit-mode commands allow you to interrogate and to reset SOS's current-line pointer. To find out where the current line is without printing it, use the Give Parameter command =. (described in Section 2.8.20). For example:

```
*=.(RET)
. IS AT 00450/5
*
```

The Move Position command (see Section 2.8.19) resets the current line. Its argument is a position specification, which can contain a line number, a page number, or their shorthand equivalents. For example, the following command moves the current-line pointer to the last line on page 3.

```
*.*/3(RET)
*
```

2.3.7 Defaults for Line and Page Numbers

One form for a complete range specification is:

```
line1/page1:line2/page2
```

You can omit various parts of the complete specification, and SOS will supply defaults for the omitted parts. The defaults discussed in this section apply only to ranges for which:

- You explicitly type a colon (:)
- You explicitly type an exclamation mark (!)
- You use the form /page

If none of these conditions holds, then the range involves a single line, and the defaults listed in Section 2.3.3 apply.

These defaults apply to all commands when you omit **part** of a range specification. For some commands, the **entire** range specification is optional. The default range that results depends on the command. See the individual command descriptions in Section 2.8 for details.

The following partial range specifications all represent legal ways to indicate a range.

1. line1/page1:line2
2. line1:line2/page2
3. /page1:line2
4. line1:/page2
5. /page1
6. :line2

Below are the complete specifications of the partial ranges listed above. The defaults SOS supplies are underlined.

1. linel/pagel:line2/pagel
2. linel/./:line2/page2
3. ^/pagel:line2/pagel
4. linel/./:*/page2
5. ^/pagel:*/pagel
6. ./.:line2/.

Table 2-2 shows the rules SOS uses to supply the omitted parts of a range specification. Note that linel defaults to ^ if pagel is supplied; but if neither is supplied, the starting position defaults to ./. (the current position). If omitted, page2 defaults to pagel, if pagel is given; if neither is given, both default to the current page.

Table 2-2
Defaults for Line and Page Numbers

If this part is omitted:	Its default is:
linel	^ (if pagel is supplied) . (if pagel is defaulted)
/pagel	/.
:	
line2	*
/page2	/pagel (whether pagel is supplied or defaulted)

Applying the rules given in Table 2-2 in some common cases gives these results:

- A command with an argument of /page acts on all lines on the given page.
- A command with an argument of :line/page acts on all lines between the current position and the given position.
- A command with an argument of :/page acts on all lines between the current line and the last line on the given page.

The examples below illustrate SOS's defaults for omitted parts of a range specification.

Command	Complete Equivalent	Meaning
*P.:80 RET	P./.:80/.	Print all lines from the current line through line 80 on the current page.
*P/. RET	P^/.:*/.	Print all lines on the current page.
P^/.: RET	P^/.:*/.	Same as above: print all lines on the current page.
P^: RET	P^/.:*/.	Same as above: print all lines on the current page.
*P:905/3 RET	P./.:905/3	Print all lines from the current line through line 905, page 3.

2.3.8 Line-Number and Page-Number Offsets

Anywhere that a line number is legal, so is an arithmetic expression forming a **line-number offset.** You can use arithmetic expressions to refer to lines whose numbers you do not know. Expressions with a plus sign (+) refer to lines later in the file, and those with a minus sign (-) refer to earlier lines.

Arithmetic expressions do not act across page boundaries, unlike the exclamation mark construction (relative number of lines). Thus, if the current line is the third line from the end of a page, the command P.+4 prints the last line on the page (as do P.+5, P.+6, and so on).

Some examples of line-number offsets follow.

*P.-2 RET	Print the line two lines before the current line.
*P^+3:. RET	Print all lines from the fourth line on the current page through the current line.
P-5!10 RET	Print the last six lines on the current page and the first four lines on the next page.
*P.-1:.+1 RET	Print three lines centered about the current line.
*P.-1!3 RET	Same as above: print three lines centered about the current line.
*P^-3 RET	Print the first line. Note that arithmetic expressions do **not** cross page boundaries.

Page-number offsets are analogous to line number offsets: anywhere that a page number is legal, so is an arithmetic expression forming a page-number offset. Some examples follow.

*P100/.+1(RET)	Print line 100 on the next page.
P305/-1!4(RET)	Print four lines starting with line 305 on the next-to-last page.
*P/.-1:/.+1(RET)	Print all text from the first line on the previous page through the last line on the following page.

You are allowed only one plus or minus sign in an arithmetic expression. Thus, P10+3 is legal, but P10+3+1 is illegal.

The syntax of line numbers and page numbers can now be formally defined. In the definitions below, l, m, p, and q are positive integers, and represent the following:

l is a line number less than 65536.

m is a line-number offset.

p is a page number.

q is a page-number offset.

A line number can consist of a number l or one of the three shorthand characters, optionally followed by an arithmetic expression. A page number has an analogous form. That is:

$$\text{line} \Rightarrow \left\{ \begin{matrix} l \\ \hat{} \\ * \\ . \end{matrix} \right\} [\pm\ m] \qquad \text{page} \Rightarrow \left\{ \begin{matrix} p \\ \hat{} \\ * \\ . \end{matrix} \right\} [\pm\ q]$$

where the braces indicate that one of the enclosed expressions must be chosen.

2.4 HOW SOS NUMBERS NEW LINES

Basically, when you insert new lines of text into a file, SOS numbers each new line by adding the current line-number increment to the previous line number. This section discusses three topics that elaborate on this basic rule:

- What the current line-number increment is, and how it is set
- How SOS behaves when an existing line intervenes before the next line number it wants to assign
- Differences in SOS's line-numbering depending on whether you are entering lines in Input mode or in Alter/insert mode

2.4.1 The Current Line-Number Increment

The current line-number increment is a parameter that SOS maintains. If you do not set it, SOS uses a value of 100 for the increment. You can find out what its value is by giving the Give Parameter command =INCREMENT(RET), and you can set it directly with the Set Parameter command /INCREMENT:n(RET).

The usual way to change the line-number increment, however, is by specifying a second argument to an Input or Replace command. With this second argument you can call for either a permanent update to the increment, or a temporary update that applies for this command only. Alternatively, you can request SOS to calculate a line-number increment that allows a known number of lines to fit in the space available to them. See the description of the Input command, Section 2.8.7, for details.

2.4.2 When an Existing Line Intervenes

SOS does not let you insert lines so that the line numbers would be out of sequence. If an existing line stands in the way of the series of new line numbers that SOS is generating, the editor eventually refuses to accept any more inserted lines since it cannot number them in sequence.

For example, suppose the current line-number increment is 100. You want to insert some new text after line 400; however, line 500 already exists. You type an Input command: I400(RET). SOS bends its usual rule (which would call for the new line to be numbered 500) and assigns the number 450, halfway in between the existing lines. SOS then accepts one input line. The next line number SOS calculates, however, is 550 (the increment, 100, plus the previous line number, 450). Again, this line number is beyond an existing line. At this point, SOS returns you to Edit mode, ending text insertion.

2.4.3 Input Mode versus Alter/Insert Mode

There are two ways in SOS to insert new lines of text: Input mode (accessed by the Input or Replace commands) and Alter/insert mode (accessed from Alter mode by the command I, or from Edit mode by the eXtend command). SOS behaves differently in the two modes when it encounters an existing line whose number intervenes in the sequence of new line numbers SOS is generating.

The difference is as follows. In Input mode, SOS lets you insert one line before an intervening line, if it is the **first inserted line.** If SOS encounters an intervening line after you have inserted one or more lines, it immediately terminates Input mode. In Alter/insert mode, upon encountering an intervening line, SOS always numbers the next line halfway between the previous line and the intervening line.

For example, suppose you want to insert text after line 100. The current increment is 100, and the next existing line is 500. If you enter Input mode to insert the text (I100(RET)), SOS numbers the new lines 200, 300, and 400; then SOS returns you to Edit mode.

However, if you enter the new lines in Alter/insert mode (A100(RET), followed by I, followed by (LF) for each new line), SOS numbers the new lines 200, 300, 400, 450, 475, 487, 493, 496, 498, and 499. At this point, no room is left for any more new lines, since the previous line and the intervening line are numbered 1 apart. SOS rings the terminal's bell and rejects further line insertion, but leaves you in Alter/insert mode.

If you have more lines to insert, you must return to Edit mode (type (RET)) and issue a reNumber (N) command.

Figure 2-2 shows the logic of SOS's line numbering. In this figure, IN is the current line-number increment; NL is the number of the new line; PL is the number of the previous line; IL is the number of a line that intervenes in the sequence of line numbers SOS is generating. In the calculation of a line number halfway between PL and IL, the notation [-1] means that SOS subtracts 1 if IL - PL is odd, but does not subtract 1 if IL - PL is even.

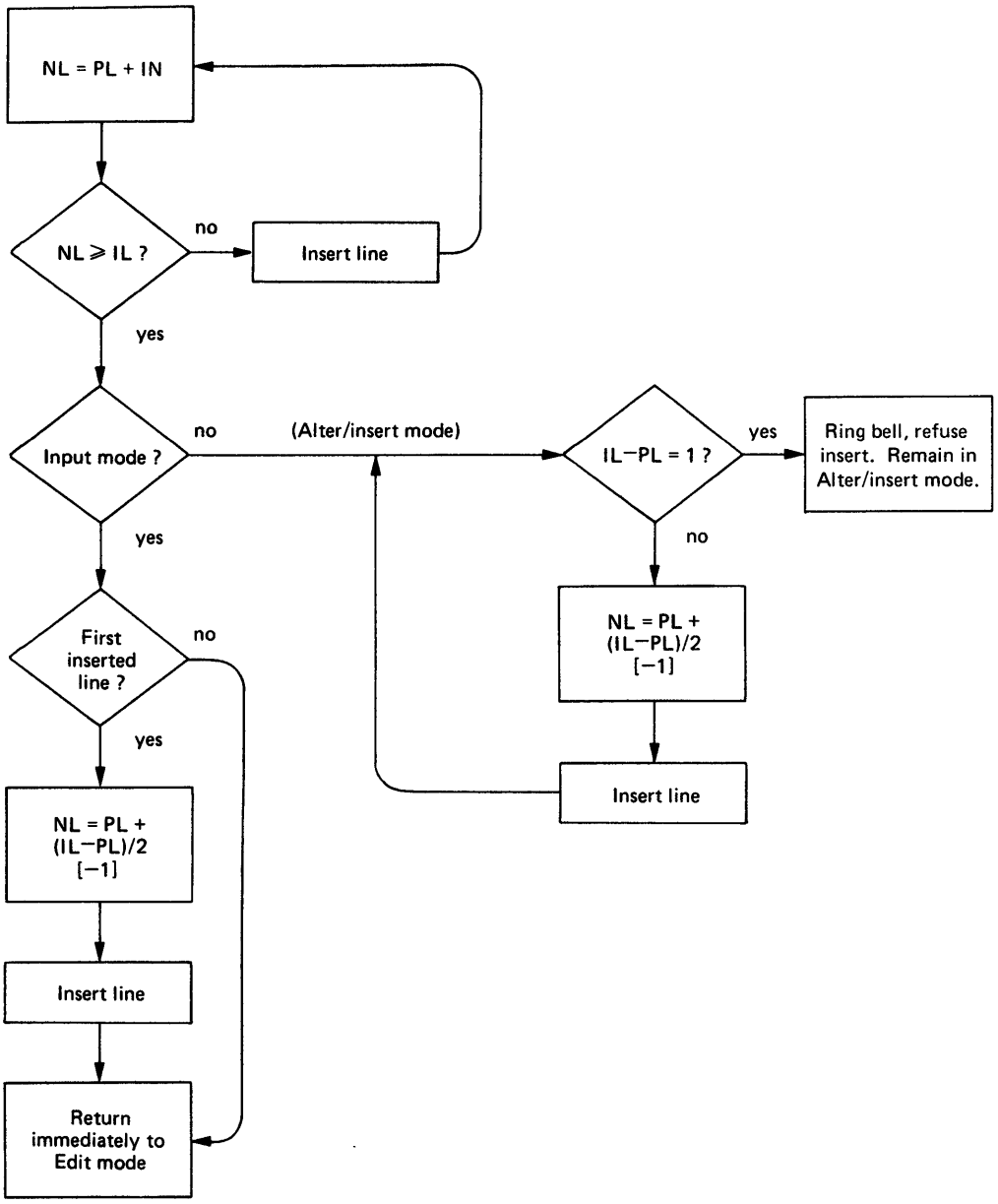


Figure 2-2 How SOS Calculates the Numbers of New, Inserted Lines

2.5 DEFAULT CONDITIONS WHEN SOS IS INITIATED

The default conditions that apply to various SOS parameters and switches when you start up the editor are listed below. The SOS parameters and switches are described fully in Section 2.9, along with a way (the symbol-assignment mechanism) to override these initial defaults.

Noexpert -- SOS starts by assuming that you are not an experienced SOS user. Therefore, it prints long, self-explanatory error messages; it asks for verification before deleting large blocks of text; and it displays deleted characters in Alter mode. All examples in this chapter assume a nonexpert user. See Section 2.9.2 for a discussion of the EXPERT switch.

Line numbering -- SOS numbers lines by 100s initially, starting at 100. When you insert lines of text, SOS uses a line-number increment of 100.

Automatic file saving -- The two parameters that govern automatic file saving (see Sections 2.9.1 and 2.7.3) are initially set to 0; therefore, SOS does not automatically save a copy of your file as you work on it. You must issue Save World commands to achieve file backup.

Lowercase characters -- Initially, SOS accepts lowercase characters as typed. If your terminal can generate lowercase characters, and if you have instructed VAX/VMS not to convert them to uppercase (using the DCL command SET), then SOS accepts and echoes them as lowercase.

2.6 LIMITATIONS AND RESTRICTIONS

Some limitations and restrictions on the use of SOS are listed below.

Line length -- No line of text can be longer than 500 characters, including spaces but excluding the (RET) that terminates the line. If you attempt to generate a line longer than 500 characters (with the Alter, Input, Join, Substitute, or eXtend commands), SOS prints the error message LINE IS TOO LONG, and returns you to Edit mode. In the case of an Alter or eXtend command, any changes that you made previously to the line are discarded, and you are returned to Edit mode.

String length -- No string used in a Find or Substitute command can exceed 200 characters in length. If you use the forms of these commands that allow multiple strings, the combined length of all such strings cannot exceed 200 characters. If you exceed these limits, SOS prints the error message SEARCH STRINGS EXCEEDED BUFFER LENGTH and returns you to Edit mode. Reissue the command with fewer or shorter strings.

Line numbers -- Line numbers cannot exceed 65535 (decimal).

2.7 USING SOS SAFELY

Some risk is associated with the use of any text editor. You can, by mistake, delete or garble large amounts of text, or even whole files. A hardware or software failure that occurs while you are using a text editor can destroy a considerable amount of work.

SOS is designed for safe use. The editor assumes that you can make mistakes; therefore (unless you have declared yourself an expert SOS user, via the EXPERT switch), SOS asks you to confirm commands that would delete large amounts of text. It also provides features that you can use to minimize the adverse effects of system malfunctions.

2.7.1 Built-in Safety Features

SOS requests confirmation before making major changes to a file. Decide mode is one safety feature of SOS; the editor lets you decide case by case on character-string substitutions, instead of making all substitutions blindly. (Section 2.8.15 describes Decide mode.)

SOS also requests confirmation before performing a Replace or Delete that would delete all the text on one page, or would delete across a page boundary.

2.7.2 The Save World Command

The SOS Save World (W) command saves "everything in the world." It creates a back-up file that is a copy of the input file containing the results of all the editing changes you have made so far.

The W command should be used frequently, perhaps every five or ten minutes in a long editing session. In this way you protect yourself against the ill effects of system failures, and against any possible errors of your own. If some system failure does occur, you lose only the work done since the last W command.

2.7.3 Two Options for Automatic File Saving

For even greater safety, you can set two SOS parameters that govern the automatic use of the Save World command. These parameters are named SAVE and ISAVE. SAVE controls automatic file saving in Edit mode; ISAVE controls it in Input mode.

To select these options, use the Set Parameter command, / (see Section 2.8.21). For example:

```
*/SAVE:10(RET)
*/ISAVE:5(RET)
*
```

SOS automatically performs a Save World command after you have issued 10 Edit-mode commands that modify the file, or after you have inserted five lines of text in Input mode, whichever comes first. While executing the save, SOS displays this message:

```
DOING AUTO-SAVE.  PLEASE WAIT.
```

You can set or reset the SAVE and ISAVE parameters at any time while SOS is operating in Edit mode. Setting either parameter to a value of 0 turns off the corresponding auto-save option. When you initiate SOS, both options are turned off.

2.7.4 Recovering Information from Temporary Files: An Example

If SOS execution is interrupted by a system malfunction, you may be able to recover some lost work from SOS temporary files. These files, and the way in which SOS uses them, are described in Section 2.13. Briefly, one or both of the files SOSxnn.TM1 and SOSxnn.TM2 may contain part of your file (x is the controller number, and nn is the number of your terminal).

The following example illustrates one way in which lost work might be recovered from an SOS temporary file. A hypothetical user, Adams, is updating a master address list on terminal number 22 when the system loses power. Not being prudent by nature, Mr. Adams has not been using the W command frequently. After this close call, however, he sets the ISAVE parameter for automatic file saving.

```
*=name@(RET)
DBA1:[ADAMS]ADDRESS.MST;17
*
 .
 .
 .
        (ESC)
        ↓
*flarkin$(RET)

PAGE 4
01050   Larkin    James    4935 Forbes Av.      Pittsburgh PA 15213
*i.,5(RET)
01055   Lawson    Adam     335 Grand Av.        Oakland    CA 94610(RET)
        Lav(ESC)(CTRL/C)(CTRL/C)(CTRL/Y)(RET)(RET)
        ↓
01060
```

Nothing works: the system has expired. Several minutes later, after Adams logs back onto the revitalized system, he recovers information from an SOS temporary file as shown below. The temporary file must be renamed, because SOS refuses to open a file it recognizes as a temporary file.

```
$ directory sosa22.*/full(RET)

DBA1:[ADAMS]
17-JUN-78  13:21

SOSA22.TM1;1          4   17-JUN-78  13:15
SOSA22.TM2;1          2   17-JUN-78  13:17
TOTAL OF 4./5. BLOCKS IN 2. FILES

$ rename sosa22.tm1 addtmp.mst(RET)

$ edit addtmp.mst(RET)

EDIT:DBA1:[ADAMS]ADDTMP.MST;1

*/isave:5(RET)
      (ESC)
*flarkin$(RET)

 PAGE 4.
01050   Larkin    James    4935 Forbes Av.      Pittsburgh PA 15213
*i.,5(RET)
 .
 .
 .
*w:address.mst(RET)
[DBA1:[ADAMS]ADDRESS.MST;18]
*
```

The exact details of the editing done so far determine how current the material contained in an SOS temporary file will be. Adams was lucky; most of the previously edited material was in the temporary file.

However, the point of emphasis here is that you cannot depend on SOS temporary files to provide reliable back-up in case of system failure. Such files are at best a last resort. The safest back-up is frequent use of the W command.

2.8 EDIT-MODE COMMANDS

This section describes in detail the forms and functions of SOS Edit-mode commands. The commands are presented in alphabetical order. The four nonalphabetic commands (. = / @) are given last. The functions of (RET) and (ESC), when they are used as Edit-mode commands, are discussed along with the Print command.

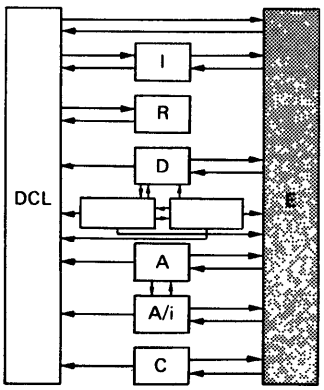


Table 2-3 summarizes the commands, their syntax, and the pages on which they are described. The following defines the terms used both in the table and in the rest of this chapter.

Definitions for the Edit-Mode Command Summary (Table 2-3)

Uppercase letters represent literals that you type verbatim. (SOS accepts either upper- or lowercase letters.) Lowercase terms represent variables, defined below. The symbol => means "is defined as." Brackets ([ ]) indicate optional parts of the argument string (except in directory or filespec, in which one set of brackets must be typed). Stacked braces ({ }) mean choose one of the enclosed expressions. Angle-brackets (< >) enclose descriptive terms. And (ESC) represents the escape key (or ALTmode, SELect, or PREfix).

file-spec => [device:][[directory]]filename[.type[;version]]

increment => <a positive integer line-number increment>

l => <a line number> (0 < l < 65536)

line => { l | ^ | * | . } [± m]

m => <an integer line-number offset>

n => <a positive integer>

p => <a page number>

page => { p | ^ | * | . } [± q]

parameter => <a legal SOS parameter or switch>

position => { line [/page] | /page }

q => <an integer page-number offset>

range => { position [{ :position | !n }] | :position | !n }

start => <a positive integer line number>

string => string1[(RET)string2[(RET)string3[(RET)string4 [(RET)string5[(RET)string6]]]]] (fewer than 200 characters)

Table 2-3
Edit-Mode Command Summary

Command	Page	Form	Arguments
Alter[1]	2-27	A	[range]
Copy[2]	2-40	C	position[=file-spec],range[,increment1[,increment2]] position=file-spec/C
Delete	2-43	D	[range]
End	2-44	E	[B] [Q] [S] [T] [:file-spec]
Overwrite input file		EB	
No output file		EQ	
Strip line numbers		ES	
No numbers, no pages		ET	
Find[3]	2-46	F	[[string](ESC)[range]] [,A] [,N] [,E] [,n] [,-]
Help	2-51	H	[:n]
Input[4]	2-52	I	[position] [,increment] [position] [;increment] [position] [;!n]
Join	2-55	J	[position]
Kill page mark	2-56	K	/page
List:LP or file	2-57	L	[range[,S] [,P[:file-spec]]] [range] [,[S] [,F:file-spec]]
Mark	2-58	M	[position]
reNumber	2-59	N	[increment] [,[range] [,start]]
Print: terminal	2-61	P	[range] [,S]
Replace[4]	2-63	R	[range] [,increment] [range] [;increment] [range] [;!n]
Substitute[5]	2-65	S	[[oldstring(ESC)newstring] (ESC) [range] [,D] [,N] [,E]]
Transfer	2-69	T	position,range[,increment1[,increment2]]
Save World	2-73	W	[B] [:file-spec]
eXtend[6]	2-74	X	[range] [,N]
Move Position	2-76	.	position
Give Parameter	2-77	=	parameter
Set Parameter	2-78	/	parameter[:n]
Command File	2-79	@	file-spec
Print next line		(RET)	
Print previous line		(ESC)	

[1] Enter Alter mode.
[2] Enter Copy-file mode (if /C).
[3] Enter Alter mode (if ,A).
[4] Enter Input mode.
[5] Enter Decide mode (if ,D).
[6] Enter Alter/insert mode.

A
Alter

2.8.1 Alter (A -- Alter Mode)

Format:

A[range]RET

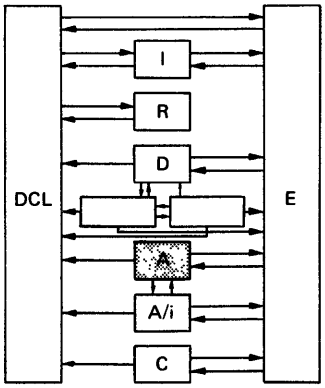
Argument:

range
Specifies the range of lines to be edited one at a time in Alter mode. If you omit range, only the current line is edited.

Current line:

After SOS completes an Alter command, it sets the current line to the last line edited.

The Alter command invokes Alter mode, a special intraline editing mode, for each line in turn in the given range. Alter mode is useful for making changes within a line without having to retype it. It is also useful for making changes involving the column positions of characters in a single line or within a range of lines.



For each line in the given range, SOS prints the line number and waits in Alter mode for your input. Eighteen commands are legal in Alter mode; any other input is ignored and rings a bell on the terminal. These commands are listed in Table 2-4 and described fully later in this section. The single-character commands are not echoed when typed. Instead, they are effected immediately. Text input that you type, replacing or adding characters within the line, is echoed as typed.

As a result, in Alter mode, you see the line taking shape as you edit it (except for text that you have backed up over or deleted).

SOS maintains an intraline pointer in Alter mode. At any given time this pointer points between two characters in the line. You can space the pointer across the line a character or a word at a time (forward or backward); you can insert text before or after the pointer position; and you can delete or replace one or more characters before or after the pointer.

As you move the pointer across a line, SOS displays characters that the pointer passes. When you move the pointer backwards across a line (to the left), characters are displayed in reverse order, surrounded by backslashes (\...\). Any characters that you delete are displayed between pairs of backslashes (\\...\\). (Note that, if you have declared yourself an expert SOS user by giving the command /EXPERT, deleted text is not echoed. See Section 2.9.2 for a description of the EXPERT switch.)

As shown in Table 2-4, many Alter-mode commands can be preceded by a positive or negative integer. For most commands, the result is to

repeat the command the given number of times. The repetition count does not make sense for some commands, and is ignored. Positive repetition counts move the pointer to the right, and negative counts move it to the left (except for DEL, for which the reverse is true).

Table 2-4
Alter-Mode Commands

Command	Meaning
B	Do not print rest of line; recycle to beginning of line with edits incorporated.
[-][n]Cx...	Change next/previous n characters to given characters.[1]
[-][n]D	Delete next/previous n characters.
E	End Alter mode for this line; do not print line.
[-][n]Itext ESC	Insert text after/before pointer; use temporary increment n for new lines.
[-][n]Kx	Kill (delete) all until nth occurrence of character x.
L	Print rest of line; recycle to beginning of line with edits incorporated.
P	Print rest of line; recycle to current position with edits incorporated.
Q	Quit Alter mode; restore original line.
[-][n]Rtext ESC	Replace next/previous n characters with inserted text.
[-][n]Sx	Skip forward/back to nth occurrence of character x.
[-][n]W	Skip forward/back n words.
[-][n]X	Delete next/previous n words.
[-][n]^	Invert case of next/previous n characters.
[-][n]<	Convert next/previous n characters to uppercase.
[-][n]>	Convert next/previous n characters to lowercase.
[-][n] BSP	Space back/forward n characters.
RET	Leave Alter mode for this line; print rest of line.
[-][n] DEL	Space back/forward n characters.
LF	Leave Alter mode for this line; print rest of line.

(continued on next page)

Table 2-4 (Cont.)
Alter-Mode Commands

Command	Meaning
[-] [n] (SP)	Space forward/back n characters.
[-] (TAB)	Skip to end/start of line.
(CTRL/R)	Do not print rest of line; recycle to current position with edits incorporated.
(CTRL/U)	Start fresh: discard edits, recycle to beginning of line, do not print line.

1. "Next" if no minus sign is typed; "previous" if minus sign is typed. The other commands that accept a minus sign are presented in similar fashion.

Figure 2-3 shows on five lines the interactive process of using Alter-mode commands to edit a single line of text. (On your terminal, you would see the single line displayed piece by piece as you gave the commands.) User input is shown in color. Alter-mode commands are not echoed as typed; rather, SOS displays their results on the text line. The arrow (↓) shows the position of the intraline pointer at the time the user types the given Alter-mode commands. The resulting printout is shown immediately below each arrow; an underline (_) marks the new position of the intraline pointer. The five Alter-mode commands used in the figure are described more fully later in this section.

If the range specified for an Alter command is empty, SOS prints the error message RANGE GIVEN DOES NOT CONTAIN ANY LINES.

Commands used in this example are:

w	moves the pointer to the right past one word and its following space(s) and/or tab(s).
i	inserts the text you type until ESC (in the example, the text is active SP).
SP	moves the pointer to the right one character at a time, and prints the character.
c	changes the following character to the next character you type (in the example, q).
RET	prints the rest of the line and ends Alter mode for this line. SOS returns to Edit mode, because the Alter command applies to only one line in the example.

```
*7.RET
10200   Each user reauiring CPU time obtains
*a.RET
        w
        ↓
10200   Each _
             iactive SP ESC
             ↓
10200   Each active _
                  w SP SP
                  ↓
10200   Each active user re_
                        cqw
                        ↓
10200   Each active user requiring _
                                   RET
                                   ↓
10200   Each active user requiring CPU time obtains
*
```

Figure 2-3 Alter-Mode Example

B (Alter mode)

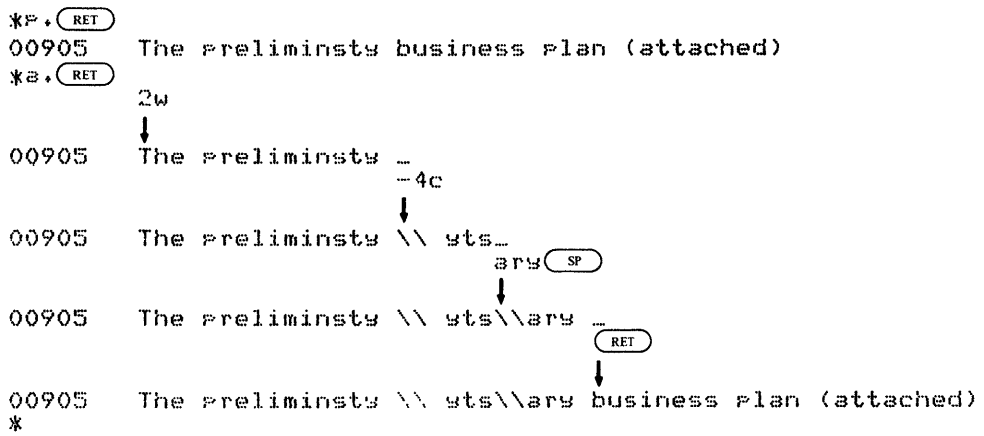
B Do not print rest of line; recycle to beginning of line

The B command tells SOS to recycle to the beginning of the line. That is, the line number is reprinted and the pointer is positioned before the first character. All edits made to the line up to this point are incorporated. SOS does not print the remainder of the line (from the current pointer position to the end) before recycling.

C (Alter mode)

[-][n]Cx... Change next/previous n characters to given characters

SOS accepts and prints the next n characters typed on the terminal. These characters replace the n characters in the line immediately to the right of the pointer. The pointer moves one character to the right as you type each character. If you type a minus sign before C, the n characters you type replace text to the left of the pointer position. SOS displays two backslashes (\\) followed by the replaced characters (in reverse order). When you type the first replacement character, SOS displays two more backslashes and echoes each replacement character as typed. The following example illustrates this sequence.



The C command is ignored if the pointer is at the end of the line (or at the beginning of the line for -n).

Three characters prematurely terminate the C command: ESC, RET, and LF. If you type nC, followed by some replacement characters (fewer than n), followed by one of these three characters, SOS terminates the C command.

- After ESC or LF, SOS is at command level in Alter mode.
- After RET, SOS continues with the next line in the range of the Alter command, or returns to Edit mode if none remain.

In each case, n characters have been deleted from the original text line.

The character DEL is ignored after a C command.

D (Alter mode)

[-][n]D Delete next/previous n characters

Typing D deletes the character that follows the pointer. More than one character is deleted if you precede the D command with a number. The deleted character(s) is (are) displayed, surrounded by pairs of backslashes: \\abc\\. If you supply a number, n, that is greater than the number of characters remaining in the line, all remaining characters are deleted. If you precede the command with a minus sign, characters before the pointer are deleted. The D command is ignored if the pointer is at the end of the line (or at the beginning of the line for -n).

E (Alter mode)

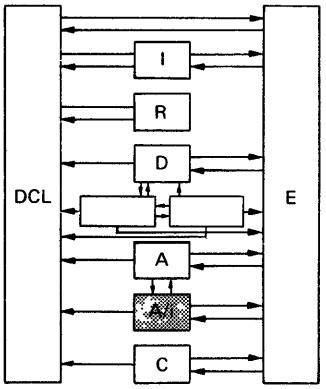
E End Alter mode for this line; do not print line

The E command tells SOS to end Alter mode for this line. The rest of the line is not printed. If more lines remain in the range you specified when issuing the Alter command, SOS proceeds in Alter mode. Otherwise, the Edit mode prompt (*) is issued. (The RET command is the usual way to exit from Alter mode; RET prints the rest of the line before returning to Edit mode, or to the next line in the range.)

I (Alter mode)

[-][n]Itext ESC Insert text after/before pointer; use temporary increment n for new lines

The I command moves SOS to Alter/insert mode. SOS accepts characters from the terminal and inserts them into the line beginning after the current pointer position. If you precede the command with a minus sign, characters are inserted before the pointer position. Alter/insert mode is terminated when you type ESC; the editor is left in Alter mode. The pointer is then positioned to the right of the inserted text, and points to the same place it did before the insertion. The inserted characters are echoed on the terminal as typed.



Both Alter/insert mode and Alter mode are terminated for this line if you type RET. In this case the final ESC is not needed. SOS prints the rest of the line and returns to Edit mode (or continues in Alter mode if more lines remain).

If you type DEL while inserting text, the previous character is deleted. DEL echoes as a pair of backslashes followed by the deleted character. Successive DELs delete and echo previous characters in turn. The next non DEL character that you type is echoed as a second pair of backslashes followed by the character. If you type more DELs than the number of characters you have inserted with the I command, SOS continues deleting characters to the left of the original point of insertion (until it encounters the beginning of the line). SOS remains in Alter/insert mode.

If you insert so much text that the overall length of the line exceeds 500 characters, SOS prints the message LINE IS TOO LONG and returns you to Edit mode. Any changes that you may have made to the line up to this point are discarded.

You can type a number before the I command. This number is not a repetition count, as it is for most Alter-mode commands. Instead, it acts as a temporary line-number increment if SOS creates a new line while inserting text. SOS creates a new line when you split an existing line by inserting LF. If you type a negative number, -n, before the I command, SOS uses +n as the temporary increment and inserts characters before the pointer position.

If you type (LF) while in Alter/insert mode, SOS creates a new line to receive the rest of the inserted text. The number of this new line is determined as follows:

- If you type a number before the I command, this number is taken as a temporary increment and is added to the current line number.
- If you do not type a number before the I command, the current increment is added to the current line number.
- If an existing line intervenes between the current line number and this new number, the new line is numbered halfway in between.
- If the next existing line is numbered 1 higher than the current line, a bell sounds and no new line is created.

After the new line is created, the pointer is positioned at the beginning of the new line; SOS is still in Alter/insert mode.

In Alter/insert mode, any character you type (with seven exceptions) is inserted into the text line. Many control characters do not have their usual effect when entered in Alter/insert mode. For example, (CTRL/S) and (CTRL/Q) have no effect on terminal output. They are simply inserted into the text line without interpretation.

The following seven characters are not treated as text in Alter/insert mode.

(ESC)	ends insertion.
(RET)	ends Alter mode for this line.
(DEL)	deletes the previous character.
(LF)	creates a new text line.
(CTRL/C)	returns SOS to Edit mode.
(CTRL/R)	retypes the line.
(CTRL/Y)	gets the attention of DCL.

K (Alter mode)

[-][n]Kx Kill (delete) all until nth occurrence of character x

This command tells SOS to delete all text from the pointer position up to, but not including, the next occurrence of the given character x. If you supply n, SOS deletes all text up to the nth occurrence of x. A minus sign causes SOS to delete backwards (to the left) from the pointer position **through** the given character. Deleted characters are printed within pairs of backslashes (in reverse order for -n).

SOS ignores the K command if the given character does not occur in the remainder of the line, or if it occurs fewer than n times.

If the EXACT switch is on (see Section 2.9.2), SOS requires an exact upper- and lowercase match with the given character x.

L (Alter mode)

L Print rest of line; recycle to beginning of line

This command tells SOS to print the rest of the line, from the pointer position onward, and recycle to the beginning of the line. The line number is reprinted and the pointer is positioned before the first character. All edits made to the line up to this point are incorporated. Compare the L command with the P command, described next (and see the following example).

P (Alter mode)

P Print rest of line; recycle to current position

The P command tells SOS to print the rest of the line, from the pointer position onward, and recycle to the current pointer position. All previous edits are incorporated. The P command lets you get a fresh look at a line you are editing. The example below shows the P command and the L command.

```
* P.(RET)
00240   while the muon sees a cross section of 0.2 barns,
* a.(RET)
         wwi
          ↓
00240   while the _
                  ^pri (DEL) (DEL) (DEL) secondary (SP) (ESC)
                  ↓
00240   while the ^pri\\irr^\\secondary _
                                        wi" (ESC)
                                        ↓
00240   while the ^pri\\irr^\\secondary muon "
                                             ^
                                             ↓
00240   while the ^pri\\irr^\\secondary muon "sees a cross section
of 0.2 barns,
00240   while the secondary muon "
                                          4 (SP) i" (ESC) 1
                                          ↓
00240   while the secondary muon "sees" a cross section of 0.2 b
arns,
00240
         iMean (ESC) 5 (SP) i, (ESC) e
          ↓
00240   Meanwhile,
* P.(RET)
00240   Meanwhile, the secondary muon "sees" a cross section of
0.2 barns,
*
```

Q (Alter mode)

Q Quit Alter mode; restore original line

The Q command immediately returns you to Edit mode. Thus it differs from the E and (RET) commands, which end Alter mode only for the current line. Q returns you to Edit mode even if more lines remain in the range you specified in the initial Alter command.

Under most circumstances, Q restores the line to its original state (as it was before editing with the Alter command). Any changes you made to the line are discarded. However, if you have formed new lines by inserting (LF), the Q command leaves them intact.

R (Alter mode)

[-][n]Rtext(ESC) Replace next/previous n characters with inserted text

The R command behaves exactly like the command [-][n]D followed by the command 0Itext(ESC). SOS deletes the next n characters after the pointer (before the pointer for -n) and enters Alter/insert mode. The characters deleted are displayed surrounded by pairs of backslashes (in reverse order for -n). (If the EXPERT switch is on, SOS does not display deleted text.) If you type (LF) after an nR command to create a new line, SOS does not use the given n to determine the new line number. In other words, the n in nR applies to the delete operation, not to the insert operation.

S (Alter mode)

[-][n]Sx Skip forward/back to nth occurrence of character x

The S command accepts one character from the terminal, without echoing it, and moves the pointer to the right until the next occurrence of that character. If you supply n, the pointer moves to the nth occurrence of the character. The pointer moves to the left past the skip character if n is negative. All characters that the pointer moves over are printed. If the pointer moves to the left, the intervening characters are printed in reverse order between single backslashes.

The character after the current pointer position is not compared to the skip character. Therefore, you can skip through a line by repeatedly typing, for example, Se. The pointer is advanced each time to the next occurrence of e.

If the EXACT switch is on (see Section 2.9.2), the S command distinguishes between uppercase and lowercase skip characters. For example:

```
*/exact(RET)
*p160(RET)
00160   Production manager Ann Shraft projects deficits of
*a.(RET)
          sa
          ↓
00160   Production m_
                    sA
                    ↓
00160   Production manager _
```

The S command is ignored if the given character does not occur in the rest of the line, or if it occurs fewer than n times. The pointer is not moved.

W (Alter mode)

[-][n]W Skip forward/back n words

This command moves the pointer to the right (to the left for -n) until it points to the first character of the next word. The pointer moves over spaces and tabs, then over the next word, then over any following spaces and tabs. If you supply n, the pointer moves to the start of the nth following word (preceding for -n).

A "word" is defined as any collection of letters and numbers.

X (Alter mode)

[-][n]X Delete next/previous n words

The X command acts like the W command, except that it deletes spaces, word(s), and following spaces through the nth word after the pointer (before the pointer for -n). SOS prints deleted characters between pairs of backslashes (in reverse order for -n). (If the EXPERT switch is on, SOS does not print deleted characters.)

Typing 99X is an easy way to delete the rest of a line.

^ (Alter mode)

[-][n]^ Invert case of next/previous n characters

This command changes the case of n characters after the pointer (before the pointer for -n). Uppercase letters become lowercase, and lowercase characters become uppercase. Nonalphabetic characters are not affected. The cursor moves past the n characters and they are echoed (for -n, the characters are echoed in reverse order between backslashes).

< (Alter mode)

[-][n]< Convert next/previous n characters to uppercase

This command converts lowercase characters to uppercase. The pointer moves past n characters (or moves back n characters if -n). All lowercase characters encountered become uppercase; and uppercase characters encountered are not affected. The n characters are echoed (for -n, in reverse order between backslashes).

> (Alter mode)

[-][n]> Convert next/previous n characters to lowercase

This command converts uppercase characters to lowercase. The pointer moves past n characters (or moves back n characters if -n). All uppercase characters encountered become lowercase; any lowercase characters encountered are not affected. The n characters are echoed (for -n, in reverse order between backslashes).

(BSP) (Alter mode)

[-][n](BSP) Space back/forward n characters

Typing (BSP) moves the intraline pointer one space to the left. The characters that the pointer passes over are printed between single backslashes. If you supply n, the pointer moves back n characters. The pointer moves forward if n is negative: -n(BSP) is exactly equivalent to n(SP). If the n (or -n) you supply is greater than the number of characters remaining, the pointer moves to the beginning (end) of the line. SOS ignores the command if the pointer is at the beginning of the line (or at the end of the line for -n).

(RET) (Alter mode)

(RET) Leave Alter mode for this line; print rest of line

This Alter-mode command is the usual way to terminate editing for the current line. (RET) ends Alter mode for this line whether you are in Alter mode or in Alter/insert mode. SOS prints the rest of the line, and either prints the line number of the next line in the range or gives the Edit-mode prompt if no lines remain.

(DEL) (Alter mode)

[-][n](DEL) Space back/forward n characters

(DEL) has the same effect as (BSP): it moves the character pointer one space to the left. (DEL) does not delete characters at Alter-mode command level, it simply moves the pointer. The characters that the pointer passes over are printed between single backslashes.

If you supply n, the pointer moves back n characters. Pointer movement is to the right if n is negative: -n(DEL) is exactly equivalent to n(SP). If the n (or -n) you supply is greater than the number of characters remaining, the pointer moves to the beginning (end) of the line. SOS ignores the command if the pointer is at the beginning of the line (or at the end of the line for -n).

(LF) (Alter mode)

(LF) Leave Alter mode for this line; print rest of line

(LF) acts like (RET) to end Alter mode for this line, if SOS is at Alter-mode command level. From within Alter/insert mode, however, (LF) signals SOS to create a new line. See the Alter-mode command I, above, for details.

(SP) (Alter mode)

[-][n](SP) Space forward/back n characters

Typing (SP) moves the pointer one character to the right and prints the character. The pointer moves n characters if you supply n. Pointer movement is to the left if n is negative; characters are printed in reverse order between single backslashes. If the n (or -n) you supply is greater than the number of characters remaining, the pointer moves to the end (beginning) of the line. SOS ignores the (SP) command if the pointer is at the end of the line (or at the beginning of the line for -n).

(TAB) (Alter mode)

[-](TAB) Skip to end/start of line

(TAB) (or (CTRL/I)) moves the pointer to the end of the line and prints characters that the pointer passes over. If you type a minus sign, SOS moves the pointer to the start of the line and prints characters in reverse order between single backslashes.

The Edit-mode command X (eXtend), which invokes Alter mode, is equivalent to the command A followed by (TAB) and I. That is, X prints the line and invokes Alter/insert mode at the end of the line.

(CTRL/R) (Alter mode)

(CTRL/R) Do not print rest of line; recycle to beginning

The (CTRL/R) command tells SOS to recycle to the current pointer position. All previous edits are incorporated. SOS does not print the remainder of the line (from the current pointer position to the end) before recycling. Thus, in Alter mode (CTRL/R) behaves the same as in Edit mode.

CTRL/U (Alter mode)

CTRL/U Start fresh: discard edits, recycle to beginning of line, do not print line

This command restarts Alter mode for the current line. Any previous changes you made are discarded. SOS immediately redisplays the line number without printing the rest of the line.

If you have formed new lines by inserting LF, the CTRL/U command leaves previous lines intact. It discards any changes made to the latest LF-created line and recycles to the beginning of that line.

CTRL/U does not have the same effect in Alter/insert mode. In this mode, CTRL/U is taken as text and is inserted into the line.

C
Copy

2.8.2 Copy (C)

Formats:

```
Cposition,range[,increment1[,increment2]](RET)
Cposition=file-spec[,range[,increment1[,increment2]]](RET)
Cposition=file-spec/C(RET)
```

Arguments:

position
: Specifies the destination for the copied lines. If the line number specified by position already exists, the copied lines are inserted after position. If not, the copied lines are inserted beginning at position.

range
: Specifies the source range of the lines to be copied. If you omit range in the second form, above, the entire file is copied.

increment1
: Specifies the line-number increment for SOS to use for the copied lines. If the source range contains page marks, increment1 is used only up to the first page mark. If you omit increment1, SOS uses the current line-number increment.

increment2
: Specifies the line-number increment for SOS to use for the lines following the final page mark in the source range. If you omit increment2, SOS uses the current line-number increment.

=
: Signals that text is to come from a second file (the copy file).

file-spec
: Specifies the copy file.

Option:

/C
: Tells SOS to enter Copy-file mode: that is, to open the copy file as a read-only file so that you can search it for the text to be copied.

Current line:

After SOS completes a Copy command, it sets the current line to the last line copied.

The Copy command inserts a copy of a block of text at a given position in the file. The block of text can be copied either from somewhere else within the file (the first usage form above) or from another file (the second and third usage forms). In the latter case, if you type /C at the end of the Copy command, SOS enters Copy-file mode so you can locate the text in the second file.

When you give a Copy command, SOS acts as if you had given an Input command at the given location, and treats each text line in the specified range as if you had typed it in.

SOS takes steps to make sure that line numbers are in proper sequence after a Copy operation. The rest of this section gives details on line numbers before and after a Copy command. For an example of how SOS handles line numbers in this situation, see Section 2.8.15, Figure 2-4.

SOS chooses a small enough line-number increment to allow all the lines in the range to fit after the destination position, using the following protocol:

1. SOS checks whether all the lines will fit if it uses the current line-number increment.

2. If the lines will not fit with the current line-number increment, SOS checks whether the lines will fit if it uses the increment you supply (increment1).

3. If the lines will not fit with increment1 (or if you have not supplied increment1), SOS tries to find a smaller increment that will make the lines fit.

SOS cannot choose a small enough increment if the number of lines to be copied is larger than the numerical difference between the line number at position and the number of the next line in the file. In this case, SOS inserts page marks. First, it copies as many lines as will fit, using either the current increment or your increment1 (if supplied), inserts a page mark, and copies the rest of the source lines with their line numbers intact. Then SOS inserts a second page mark.

SOS tells you what increment it has used with the message INC1=n. If it inserts a page mark, SOS issues the message INC1=ORDER n AND P/M INSERTED.

If the range of lines to be copied already contains page marks, SOS renumbers the copied lines only up to the first page mark. The page mark is copied, and all the lines up to the last page mark in the range are copied with their line numbers intact.

For the lines after the last page mark, SOS follows a protocol similar to the one it uses for lines before the first page mark:

1. If the lines will fit, SOS copies them using the default line-number increment.

2. If the lines will not fit with the default line-number increment, SOS uses the increment you specify (increment2).

3. If the lines will not fit with increment2 (or if you have not supplied increment2), SOS chooses a smaller increment.

If SOS cannot choose a small enough increment, it inserts another page mark, copies the lines with numbers intact, and inserts a final page mark.

If the source range contains page marks, SOS tells you the increment it uses for the lines copied after the final page mark. The message is INC2=n if SOS does not have to insert a final page mark, and INC2=ORDER n AND P/M INSERTED if it does insert one.

If the specification of the source range contains the shorthand symbol ^, meaning the first line on a page, SOS copies the page mark that precedes the first line before it begins to copy the text. If the range specification contains *, meaning the last line, SOS copies the following page mark as well.

Note that SOS copies the page mark(s) whether you specify the symbol(s) ^ and/or *, or whether SOS supplies them as defaults. Thus, SOS copies the preceding page mark if you type the range

```
/3:50/4
```

This is because the initial line number for page 3 defaults to ^. SOS copies the succeeding page mark if you type the range

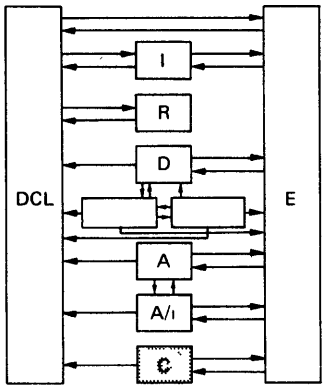
```
20/3:/4
```

This is because the final line number for page 4 defaults to *. If you type only a single page number for the range, SOS copies both the preceding and the succeeding page marks for that page. For example:

```
/5
```

This is because SOS expands /5 to ^/5:*/5.

2.8.2.1 **Copy-file Mode** - This mode allows you not only to copy a block of text from another file, but also to search the copy file to locate the text. You call for Copy-file mode by using the Copy command in the third format given above. For example:




```
*C455/3=CONTROL.OLB/C RET
```

The equal sign (=) means that text is to be copied from a second file, and the final /C signals Copy-file mode. If you do not type /C (that is, if you use the second usage form given above), the entire file is copied at the point specified. In this example, the file would be copied at or after line 455, page 3, depending on whether or not line 455 exists.

If you type the final /C, SOS prompts C* and waits for you to type a Find, Print, List, or Move Position command. Only a limited number of commands are legal in Copy-file mode because the copy file is read-only while it is open. The Edit-mode commands RET and ESC are legal, because they are essentially abbreviations of the Print command. Also legal, but of limited utility, are Give Parameter, Set Parameter, and Command File. Any other command results in the message COMMAND INVALID FOR READ ONLY FILE.

After you locate the desired lines in the copy file, type

```
*E RET
```

SOS responds with

```
ENTER RANGE OF LINES:
```

At this point you complete the Copy command with the range and, if desired, one or two line-number increments. You can use line-number abbreviations (. ^ *) in specifying the range. The values of the current line, current page, last line, and so forth refer to the copy file, not to the original file. SOS completes the copy operation and responds with * (the Edit-mode prompt).

D
Delete

2.8.3 Delete (D)

Format:

D[range]RET

Argument:

range
Specifies the lines to be deleted. If you omit the range, the default is to delete the current line.

Current line:

After SOS completes a Delete command, it sets the current line to the last line deleted (which no longer exists).

This command deletes the lines in the specified range. If no lines are in the range, SOS prints the error message RANGE GIVEN DOES NOT CONTAIN ANY LINES.

Before deleting all lines on a page, or deleting across a page boundary, SOS requests confirmation with one of the following questions:

```
DELETE ENTIRE PAGE? Y OR N:
DELETE ACROSS PAGES? Y OR N:
```

SOS then waits for you to type either Y or N (a RET is not needed). SOS rejects any character other than Y, y, N, or n and repeats the question.

After deleting the lines, SOS reports the results of the deletion with a message such as the following:

```
31 LINE(S) DELETED (00145/2:00250/4).
*
```

If the EXPERT switch is on, SOS neither requests confirmation nor tells you how many lines were deleted.

E
End

2.8.4 End (E)

Formats:

E[:file-spec]RET
EB[:file-spec]RET
EQ[:file-spec]RET
ES[:file-spec]RET
ET[:file-spec]RET

Arguments:

file-spec
: If given, specifies the output file. If omitted, SOS uses the current output file specification.

B
: Tells SOS not to create a "back-up" file. The output file has the same version number as the input file, and the input file is effectively deleted.

Q
: Tells SOS to quit without writing an output file.

S
: Tells SOS to write the output file without line numbers, but with page marks.

T
: Tells SOS to write the output file without line numbers or page marks.

Current line:

Not applicable: after the End command, SOS terminates.

The End command terminates the editing session and returns you to DCL. Different versions of the End command allow you to exit with or without a back-up file, line numbers, or page marks.

SOS tells you the specification of the output file it writes. It gives no specification if you exit without an output file (EQ). If you have not changed the file since it was last written, SOS appends the note (NO CHANGES) to the file specification, and exits without rewriting the output file. If you do not give a file specification with the E command, SOS writes an output file with the same file specification as given in the last Save World command. If no Save World was given, then the file name and type are the same as those of the input file, but the version number is higher by 1. If you supply a file specification, the output file is written with that specification.

When supplying a file specification, you cannot make SOS write its output file over an existing file. If a file already exists with the same name and type as the given file specification, SOS increments the version number before writing the output file. If the given file specification is identical to that of an existing file, including the version number, SOS rejects the command with the message FILE ALREADY EXISTS and requests a new filename with the prompt ENTER NEW FILENAME:.

Use the EB form of the End command to request SOS not to create a back-up file when it exits. The version number of the output file is not incremented. Thus, the information in the original file is lost, unless you have previously given a Save World command that incremented the version number.

The ES form requests SOS to write the output file without line numbers. You can use ES to save disk space, since files without line numbers require less storage space. However, SOS requires significantly more processing time to generate a file without line numbers. This is because of the way SOS handles temporary files; see Section 2.13.1.

On VAX/VMS systems, you need not use ES to make allowances for programs that cannot deal with line-numbered files. (This is desirable on some other systems with other implementations of SOS.) On VAX/VMS, the input/output system transparently handles line-numbered files. Therefore, the use of ES should be rare.

Use the ET form of the End command to clean up a file that has become messy with uneven line numbers and with page marks scattered randomly throughout. ET tells SOS to write the output file as a **textual file**, without page marks as well as without line numbers. The next time SOS opens the file, it renumbers it. You can use the symbol-assignment mechanism to control SOS's line-numbering defaults (see Section 2.9.3). Thus, you can start fresh with the file.

One use of the ET form of the End command is in preparing command files to serve as command input to SLP, the batch-oriented editor (see Chapter 3).

You can use any combination of the B, S, and T options with the End command, in any order. Some useful forms are given below.

ESB(RET) Write the output file without line numbers. Do not increment its version number.

EBT(RET) Write the output file without line numbers or page marks. Do not increment its version number.

F
Find

2.8.5 Find (F)

Formats:

```
F[[string](ESC)[range][,A][,N][,E][,n][,-]](RET)
F[[string1(RET)
   .
   .
  string6](ESC)[range][,A][,N][,E][,n]](RET)
```

Arguments:

string
: Specifies the string of characters (fewer than 200) that SOS is to search for. SOS prints the first line in which the string is found. If you omit string, SOS uses the current find string.

range
: Specifies the range of lines to be searched. If you omit range, SOS searches from the line after the current line to the end of the file.

string1...string6
: Specifies six or fewer distinct strings that SOS is to search for. The total number of characters in the strings may not exceed 200.

Options:

,A
: Tells SOS to enter Alter mode automatically when the string is found. The intraline pointer points before the beginning of the string in the line.

,N
: Tells SOS to print only the line number(s) of the line(s) in which the string is found.

,E
: Tells SOS not to treat uppercase and lowercase letters as equivalent in the search string.

,n
: Tells SOS to find the next n occurrences of the string in the given range.

,-
: Tells SOS to print the next line that does **not** contain string.

Current line:

After SOS completes a Find command, it sets the current line to the last line found; however, if the search fails, the current line is unchanged.

Common short forms:

F (RET)
Search for the string(s) specified in the previous complete Find command, starting at the line after the current line, and continuing to the end of the previously specified range.

(ESC)
↓
F$ (RET)
Search for the previously specified string(s), starting after the current line and continuing until the end of the file.

Use the Find command to find a given string of text occurring within a given range of lines. SOS prints the first line containing the string within the range. If the range includes more than one page, and SOS finds the string on a page other than the current one, it prints PAGE n. before printing the line containing the string. If SOS cannot find the string in the given range, it prints STRING NOT FOUND, SEARCH FAILED. The current line is unchanged.

To find the first occurrence of a word on a given page, use the following form:

(ESC)
↓
Fisomer$/4 (RET)

This form of the command means find and print the first line on page 4 that contains the string "isomer".

To locate the next occurrence of the same string you gave in the previous Find command, use only the range argument:

(ESC)
↓
*F$/6 (RET)

This form means find and print the first line on page 6 that contains the string "isomer" (from the previous Find command). If you use this form of the Find command before you issue a complete Find command, SOS prints ILLEGAL SEARCH STRING GIVEN.

If you omit the range specification, SOS begins searching at the line following the current line, and continues through the end of the file. For example:

(ESC)
↓
fdeoxyribo$ (RET)

This form means find the first occurrence of the string "deoxyribo" beginning with the line following the current line through the end of the file.

You can give the Find command repeatedly, each time finding the next occurrence of a string in the file. To do this, simply type F(RET). For example:

```
          (ESC)
           ↓
*frandom$/1:/*(RET)
00320  but random perturbations of the lattice structure
*f(RET)
01310  a certain degree of randomness in the process.
*f(RET)

 PAGE 2.
00110  a pseudo-random number in the interval [0,1), chosen
*f(RET)
STRING NOT FOUND, SEARCH FAILED
*=.(RET)
. IS AT 00110/2
*
```

You can search for as many as six different strings at once by using a special form of the Find command. Instead of typing (ESC) after the first string, type (RET); SOS prompts you with F* for another search string. After typing all the search strings in this way, type (ESC) and the range and options (if desired).

The multiple search strings can contain up to 200 characters in all.

An example using two search strings is given below. Note that the user has typed ,99 as an option at the end of the Find command. SOS finds the first 99 (or fewer) occurrences of either search string in the range.

```
*frandom(RET)
          (ESC)
           ↓
F*stochastic$/1:/*,99(RET)
00320   but random perturbations of the lattice structure
00760   from the stochastic equation (Eq. 4-2).
01310   a certain degree of randomness in the process.

 PAGE 2.
00110   a pseudorandom number in the interval [0,1), chosen

 PAGE 4.
01210   in the study of stochastic processes.
*
```

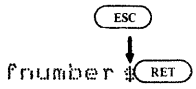
A common error in using the Find command is to type (RET) after a search string when you meant to type (ESC). SOS assumes you intend to specify multiple search strings, and prompts F*. At this point, you can type (ESC) and complete the Find command normally.

2.8.5.1 **Find Options** - The complete Find command, including options, is shown as the first format given at the beginning of Section 2.8.5. You can supply options in any combination and in any order.

If you give a Find command with one or more options, these options will still be in effect if you later use the abbreviated form F(RET). The ,n option is an exception: it applies only to the complete Find command.

To reset the options while still using an abbreviated Find command, type F(ESC)(RET).

If you do not use the ,E option, the command below would indiscriminately find a line containing any of the following: number, Number, or NUMBER.



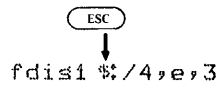
This ambiguity can be useful: for example, when you want to find a word whether it begins a sentence or not. But if you want SOS to distinguish uppercase from lowercase characters when searching, use the ,E option.

Using the ,E option is equivalent to turning on the EXACT switch (see Section 2.9.2) for Find commands only.

Use a large number like 999 for the ,n option to find all the strings in a given range. Unlike the other three options, the ,n option applies only to one execution of the Find command. For example, if you give the command FFIRST(ESC),2(RET) and later use F(RET), SOS finds only one more occurrence of the string FIRST.

If you use the ,A (Alter mode) and ,- (do not find) options in the same Find command, SOS enters Alter mode with the intraline pointer positioned at the beginning of the line that was found.

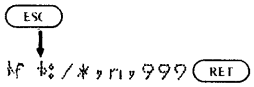
Examples:



Print the first three lines, from the current line through the last line on page 4, that contain the string digi. Do not match a string unless its case (upper or lower) agrees exactly with that of the given string.

f$ (RET)

Print the next line after the current line that contains the string digi. The search starts at the line after the current line and continues to the end of the range last specified (to the end of page 4). An exact pattern match is still required. SOS prints only one matched line; the ,3 option specified in the previous Find command applies only to that one command.



Print only the line numbers (and page numbers) of the lines in the rest of the file that contain the string digi. Exact pattern matching is no longer called for. Consequently, SOS matches Digi and DIGI, for example, as well as digi.

```
*=string RET
  FIND=
disi
  SUBSTITUTE=
  FOR= *
```

Display the strings currently in effect for the Find and Substitute commands. See Sections 2.9.1 and 2.8.19, respectively, for details on the STRING parameter and the Give Parameter command (=).

```
*fMunich RET
F*Bonn RET
       ESC
F*Berlin$.;/*,a,e,999 RET
```

Find all occurrences of any of the strings Munich, Bonn, and Berlin in the rest of the file. Do not match a string unless it starts with an uppercase letter. As each string is found, enter Alter mode for that line, with the intraline pointer positioned before the start of the string that was found.

H
Help

2.8.6 Help (H)

Format:

H[:n]RET

Argument:

n
Specifies the number of the module in the help package that SOS is to start printing.

Current line:

The Help command does not affect the current line.

The Help command prints a help package on your terminal. This package is composed of a number of modules. A control feature lets you skim quickly through the modules.

After the introductory module, the help package contains summaries of SOS features and functions similar to Appendix A, but formatted for printing on a terminal.

If you type the Help command with no argument, SOS prints the entire help package. The introductory module contains a list of the other modules and their numbers, plus instructions for printing the help package on the line printer.

If you type CTRL/O while a module is printing, SOS skips to the beginning of the next module. You can skim through the help package by typing successive CTRL/Os. Typing CTRL/C returns you immediately to Edit mode.

If you type the Help command with an argument n, SOS prints module number n and continues to the end of the help package.

As an example of how you can use the Help command, suppose you want to see a list of Alter-mode commands. Proceed as follows:

1. Type HRET. SOS prints the introductory module.
2. Type CTRL/C when you read that module number 6 summarizes Alter-mode commands. SOS prints the Edit-mode prompt *.
3. Type H:6RET. SOS prints module number 6.
4. Type CTRL/C when SOS begins to print module number 7. SOS prompts with *.

I
Input

2.8.7 Input (I -- Input Mode)

Formats:

I[position][,increment]RET
I[position][;increment]RET
I[position][;!n]RET

Arguments:

position
Specifies the destination for the inserted text; position is the line number of the first inserted line, if a line does not already exist at position. If a line exists at position, the line number is set to position plus the current line-number increment (or the specified increment). If another line exists at or before this new line number, the new line is numbered halfway in between. If you omit the position argument, insertion continues at the position where the last Input command left off.

increment
Specifies the new current line-number increment, if preceded by a comma. If preceded by a semicolon, the argument is a temporary increment only for this insert. If you omit increment, the current line-number increment is used.

n
Specifies the number of lines you want to insert. SOS calculates an appropriate temporary line-number increment.

Current line:

After SOS completes an Input command, it sets the current line to the last line inserted.

Common short forms:

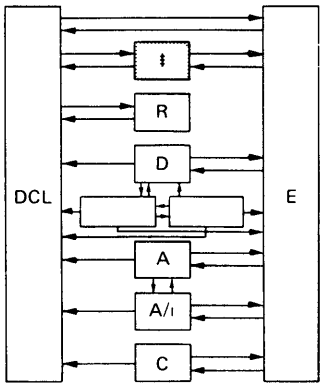
I.RET
Insert after the current line.

I*RET
Insert after the last line on this page.

I/*+1RET
Start a new page at the end of the file and begin inserting there.

IRET
Continue inserting after the line most recently inserted, using the line-number increment given in the last complete Input command. You must issue a complete Input command before using this form.

Use the Input command to enter new lines into a file. Its arguments are a single position (line/page) and/or a line-number increment. Text insertion begins (SOS enters Input mode) at or after the line number specified by position. SOS accepts successive lines of text from the terminal, and inserts them into the file. Successive line numbers are determined by adding the current increment to the previous line number.



The increment argument, if specified with a comma, becomes the new line-number increment. If specified with a semicolon, increment acts as the line-number increment for this command only. Subsequent commands continue to use the default increment (except I(RET), which uses the temporary increment).

Use the third form of the Input command when you know exactly how many lines you want to insert. In this case, SOS calculates a line-number increment that allows the given number of lines to fit before the next existing line.

Exit Input mode by typing (ESC). If you type (ESC) at the beginning of a new input line, that (empty) line is not inserted into the file. The last full line inserted becomes the current line. You can also return to Edit mode by typing (ESC) at the end of the last line to be inserted.

In some circumstances, SOS returns you to Edit mode automatically. These circumstances are given below, along with ways to recover and continue inserting text:

- A line exists whose number is equal to or less than the number that SOS calculates for the next inserted line, and this line is not the first line inserted. (To continue inserting text, issue another Input command, specifying a smaller increment.)
- The line number that SOS calculates for the next inserted line is greater than 65535. (Renumber part of the file.)
- You try to insert a line longer than 500 characters. (Use shorter lines.)

After completing an Input command, when you have returned to Edit mode, SOS remembers the line number and page number of the next line to be inserted. If you type I(RET), text insertion resumes at this point. SOS uses the increment specified in the last complete Input command you typed, even if it was a temporary increment. Thus, in this special case, the temporary increment endures after the Input command in which it was established.

If you specify an existing line as position in an Input command, and this line is followed by another line with a number higher by 1, SOS prints the error message INSUFFICIENT LINE NUMBERS FOR INSERTION. In this case, use the reNumber command (N), then reissue the Input command.

Example:

```
*P.!2(RET)
00130    MOV       R5,CNTBL(R3)       ;SAVE UCB POINTER
00140    CLR       U.CW2(R5)          ;CLEAR ALL SWITCHES
*I.,2(RET)
00142    BEQ       10$                ;IF EQUAL YES(RET)
00144    MOV       I.TCB(R1),R0       ;GET REQUESTOR TCB ADDRESS(RET)
        (ESC)
          ↓
00146    $
*(RET)
00150    MOV        170.,U.CNT(R5)    ;SET COUNT FOR 170 NULLS
*I(RET)
                                                                  (ESC)
                                                                    ↓
00146    BNE       65$                ;IF N.E. DON'T PUNCH TRAILER
*=BIG(RET)
4
*I*/4(RET)
00430    VERSION 21A, 2-21-78. GNL.(RET)
        (ESC)
          ↓
00432    $
*E(RET)
[DBA1:[LANEY]PPDRV.MAC;18]

$
```

J
Join

2.8.8 Join (J)

Format:

J[position]RET

Argument:

position
Specifies the line to be joined to the following line. If you omit position, the current line is joined to the following line.

Current line:

After SOS completes a Join command, it sets the current line to the new line made from the two old lines; that is, the current line number becomes the specified position.

The Join command joins two lines together. The joined line is given the number of the first line of the pair, that is, of the line at the specified position. It becomes the current line after the Join command completes. The line number that previously belonged to the second line of the pair ceases to exist.

SOS refuses to join two lines if the resulting line would be longer than 500 characters. It prints the error message LINE IS TOO LONG.

If the line specified by position is the last line on a page, SOS prints ILLEGAL SYNTAX OF COMMAND.

Example:

```
*P400!3RET
00400   The goal is always

 PAGE 2.
00610   to write clear, unambiguous
00650   functional specifications that are easy to understand.
*J400/1RET
ILLEGAL SYNTAX OF COMMAND
*k/2RET
*J400RET
*P.RET
00400   The goal is alwaysto write clear, unambiguous
*a.RET
        %s SP i SP ESC e
        ↓
00400   The goal is always
*P.RET
00400   The goal is always to write clear, unambiguous
*
```

K
Kill Page Mark

2.8.9 Kill Page Mark (K)

Format:

K/page RET

Argument:

/page
Specifies the page mark to be deleted.

Current line:

The Kill Page Mark command does not affect the current line.

This command deletes ("kills") a page mark that you have inserted with the Mark command, or that SOS has inserted in the course of a Copy or Transfer command. When a page mark is deleted, the text that was previously on that page appears at the end of the preceding page. (A page mark is considered to be located at the head of its page.)

You can delete only one page mark at a time.

Deleting a page mark can leave line numbers out of order. SOS warns you with the message LINES OUT OF ORDER. In this case, you must immediately renumber the current page using the reNumber command.

If you try to delete a nonexistent page number, SOS prints the message PAGE DOES NOT EXIST. If you try to delete page mark 1 (K/1), SOS responds CANNOT KILL INITIAL PAGE MARK. (SOS considers the text immediately after the start of the file to be on page 1, but it does not insert a physical page mark there.)

L
List

2.8.10 List (L)

Formats:

L[range][,[S][,P[:file-spec]]]RET
L[range][,[S][,F:file-spec]]RET

Argument:

range
Specifies the range of lines to list. If you omit range, SOS lists the entire file.

Options:

,S
Tells SOS to suppress line numbers in the listing.

,P:file-spec
Tells SOS to produce a disk file, with the given file specification, formatted for later printing.

,F:file-spec
Tells SOS to produce a disk file formatted as an ordinary SOS text file.

Current line:

After SOS completes a List command, it sets the current line to the last line listed.

The List command allows you to generate either a printer listing or a disk file containing all or part of the file being edited. The List command has an advantage over the DCL PRINT command for printing SOS files. The List command generates headings on each page of printer output giving the SOS page number, as well as each listing page's sequential number within the SOS page.

The heading contains the specification of the file being listed; the date and time printed; and a page number in the form m-n, where m represents the page number in the SOS file, and n is the sequential number of the listing page within the SOS page. (When n=1, the printing of "-n" is suppressed.). These headings are generated if you specify ,P (whether the output goes to a printer or to a disk file). The headings are also generated if neither ,P nor ,F is specified.

The SOS parameter LENGTH controls the page length of listings produced by the List command, ,P option. Its initial value is 55 lines. You can reset this parameter if desired (see Section 2.9.1).

You can specify a particular printer by using the ,P option in the form ,P:LPan: (a is the controller designation and n is the printer number). LPan: is a legal file specification. If you omit file-spec from the ,P option, or if you supply neither ,P nor ,F, the listing is sent to LPA0.

The option ,F:file-spec generates a disk file containing the lines in the given range, but without the page headings described above. The disk file thus produced is a normal SOS file. Any page marks in the range are included in the file. The ,S option, if present, tells SOS to suppress line numbers in the disk file.

M
Mark

2.8.11 Mark (M)

Format:

M[position]RET

Argument:

position
Specifies the position (line/page) that is to be the first line on a new page. SOS inserts a page mark before the given position. If you omit position, the current line is assumed.

Current line:

After SOS completes a Mark command, it sets the current line to the first line on the new page.

The Mark command inserts a page mark at a specified point in the text and renumbers all subsequent pages.

If the line number in the position specification does not exist, SOS inserts the page mark immediately after the previous, lower-numbered line.

If you give the Mark command for the first line on a page, SOS inserts a page mark immediately after the existing page mark. The page that previously existed is then empty.

After the completion of a Mark command, the current line is the first line on the new page. Therefore, if you give the Mark command shown below, the current line becomes 300/4.

```
* M300/3RET
```

The new line is the same line that you referred to in the command as 300/3, but it is now on page 4.

2.8.12 reNumber (N)

Format:

```
N[increment][,[range][,start]]RET
NA[increment][,[range][,start]]RET
NP[increment][,[range][,start]]RET
```

Arguments:

increment
: Specifies the line-number increment between the renumbered lines. If you omit it, SOS uses the current line-number increment.

range
: Specifies the range of lines to renumber. If you omit it, SOS renumbers the entire file.

start
: Specifies the starting line number for the renumbering. If you omit it, SOS starts renumbering at the value of increment.

A
: Tells SOS to add increment to every line number in the given range.

P
: Tells SOS to renumber all lines in the range sequentially. SOS does not reset the starting line number at each page boundary.

Current line:

After SOS completes a reNumber command, it sets the current line to the last line renumbered.

This command renumbers the lines in the given range. You must use reNumber whenever SOS prints the message LINES OUT OF ORDER. More commonly, you can renumber upon receiving the error message INSUFFICIENT LINE NUMBERS FOR INSERTION (or at any other time).

The first argument, increment, specifies the increment between line numbers. The first line renumbered is given this number, unless you specify the third argument (start). Each succeeding line is given a number that is the sum of the increment and the previously assigned number.

If SOS crosses a page mark while renumbering, it assigns the increment number to the first line on the new page (unless you specify NP). For example, you type the following command:

```
*N10,/2/3RET
```

Before renumbering, the line numbers on pages 2 and 3 might be as shown below.

```
Page 2     115   126   137   150
Page 3     900   911   952  1022
```

After renumbering, the line numbers are:

```
Page 2      10    20    30    40
Page 3      10    20    30    40
```

If you use the NP form of the command, SOS does not reset its numbering when it encounters a page boundary. The entire range is numbered in ascending order. Thus, using NP in the above example produces:

```
*NP10,/2+3 RET

Page 2      10    20    30    40
Page 3      50    60    70    80
```

The NA form of the reNumber command tells SOS to add the specified increment to each line number in the range. If you use NA, the start argument is meaningless. If supplied, it is ignored.

If you do not specify a range, SOS renumbers the entire file. If the range is empty, SOS prints RANGE GIVEN DOES NOT CONTAIN ANY LINES.

If, while renumbering, SOS assigns a line number larger than 65535, it stops renumbering immediately and prints the message LINE NUMBER STEP TOO LARGE - LINES OUT OF ORDER. At this point, you must reissue the reNumber command using a smaller increment or a smaller starting number.

If, after issuing a Kill Page Mark (K) command, you get the error message LINES OUT OF ORDER, renumber using /. as the range (the current page).

P
Print

2.8.13 Print (P)

Format:

P[range][,S]RET

Argument:

range
Specifies the range of lines to print. If you omit range, SOS prints 16 lines (or the number of lines given by the parameter PLINES).

Option:

S
Tells SOS to suppress line numbers and to print only the text.

Current line:

After SOS completes a Print command, it sets the current line to the last line printed.

Common short forms:

P RET
Print 16 lines (or the number of lines given by PLINES) starting at the current line; cross page boundaries if necessary.

P/. RET
Print all lines on the current page.

RET
Print the next line.

ESC
Print the previous line.

The Print command prints on your terminal the lines in a given range, including line numbers. If the range includes a page boundary, SOS prints a blank line and PAGE n. to mark it.

Use the option ,S to suppress line numbers in the Print command. SOS still prints PAGE n. when it crosses a page boundary.

You can interrupt typeout at any time by typing CTRL/S. Output pauses until you type CTRL/Q to resume where it left off. To suppress all remaining output from a Print command, type CTRL/O. There may be a slight pause before SOS gives the Edit-mode prompt * because SOS continues to move output to a buffer area after you type CTRL/O. If you type a second CTRL/O during this pause, typeout resumes at some later point in the output.

If you use the Print command without arguments (P RET), SOS prints a number of lines (initially 16) beginning with the current line. You can change the number of lines printed by P RET by resetting the SOS parameter PLINES. The command is:

```
*/PLINES:n RET
*
```

See Section 2.8.21 for details on the Set Parameter command /.

The Edit-mode commands (RET) and (ESC) act as abbreviated forms of the Print command. They tell SOS to print, respectively, the next line and the previous line in the file. With (RET) and (ESC), you can step through a file a line at a time in either direction.

(RET) and (ESC) operate regardless of page boundaries. For example, suppose the current line is the last line on page 3 when you type (RET):

```
* =.(RET)
. IS AT 1600/3
* (RET)

 PAGE 4.
00100   Guidelines for creating RUNOFF files online are
*
```

These two special abbreviations of the Print command can be defined as follows:

$$\text{(RET)} \Rightarrow \begin{cases} \text{P.+1} & \text{if } . \neq * \\ \text{P\^{}/.+1} & \text{if } . = * \end{cases}$$

$$\text{(ESC)} \Rightarrow \begin{cases} \text{P.-1} & \text{if } . \neq \text{\^{}} \\ \text{P*/.-1} & \text{if } . = \text{\^{}} \end{cases}$$

Example:

```
* P.-1:3(RET)
00530   trade-off involved in using relatively organized
00540   files instead of sequentially organized ones: in
00545   most cases, retrieval time is reduced significantly.
* (RET)
00550   However, there may be unused space on the file.
* (RET)

 PAGE 3.
00010   Access Methods:
* P*(RET)
00660   VAX-11 RMS file organizations listed in Table 4-1.
* =.(RET)
. IS AT 00600/3
* =BIG(RET)
3
* (RET)
NO SUCH LINE EXISTS
*
```

R
Replace

2.8.14 Replace (R)

Formats:

R[range][,increment]RET
R[range][;increment]RET
R[range][;!n]RET

Arguments:

range
Specifies the range of lines to be deleted and replaced with new text. SOS enters Input mode. The first new line of text is given the number of the first line in the range. If you omit range, the current line is replaced.

increment
Specifies the new current line increment, if preceded by a comma. If preceded by a semicolon, it is a temporary increment for this command only. If you omit increment, the current line-number increment is used.

n
Specifies the number of lines you want to insert in place of the deleted range. SOS calculates an appropriate line-number increment.

Current line:

After SOS completes a Replace command, it sets the current line to the last line inserted.

Common short forms:

R.RET
Replace the current line with a new line or lines typed in.

RRET
Replace the current line.

The Replace command acts like a Delete command for the given range, followed by an Input command at the first line in the range. SOS tells you how many lines have been deleted (unless the EXPERT switch is on) and then enters Input mode. You can optionally supply an increment for the Replace command, as you can for the Input command.

The increment, if supplied, acts as either a temporary (;) or permanent (,) update of the current line-number increment. That is, use the ,increment argument to change the default line-number increment. Use the ;increment argument to define a temporary increment for this command only. Subsequent Input and Replace commands continue to use the default increment.

Use the third form of the Replace command, Rrange;!n, if you know exactly how many lines you have to insert. SOS calculates an appropriate (temporary) line-number increment.

The line number of the first inserted line is that of the first line that SOS deleted in the range. Subsequent line numbers are determined by adding the current increment to the previous line number.

If you specify a range that crosses page boundaries, or that includes an entire page, SOS asks for confirmation before deleting anything. SOS prints one of the following queries:

```
DELETE ACROSS PAGES? Y OR N:
REPLACE ENTIRE PAGE? Y OR N:
```

SOS does not request this confirmation before deleting if the EXPERT switch is on.

In only one particular does the Replace command differ from an equivalent combination of Delete and Input commands. The Replace command does not define the proper parameters to allow the abbreviated Input command I(RET). You can use this abbreviated form only after typing a full Input command.

Examples:

```
* R/2,50(RET)
REPLACE ENTIRE PAGE? Y OR N:Y
```

Replace all lines on page 2 and insert new lines, beginning with the first line deleted. Use an increment of 50; make 50 the current line-number increment.

```
* R150;!2(RET)
```

Replace line 150 on the current page with two lines. SOS calculates a temporary increment that allows the two lines to fit.

```
* R337!5;!7(RET)
```

Replace the five lines starting at line 337 with seven new lines.

S
Substitute

2.8.15 Substitute (S)

Formats:

```
S[[oldstring(ESC)newstring](ESC)[range][,D][,N][,E]](RET)
S[oldstring1(RET)
   .
   .
  oldstring6(ESC)newstring1(RET)
   .
   .
  newstring6(ESC)][range][,D][,N][,E](RET)
```

Arguments:

oldstring
: Specifies any string of characters (fewer than 200).

newstring
: Specifies a string of characters that SOS is to substitute for all occurrences of oldstring within the range.

range
: Specifies the range of lines over which SOS is to make the substitution. If you omit range, SOS performs the substitution only on the first match found, starting its search with the current line, proceeding if necessary to the end of the file.

oldstring1...oldstring6
: Specifies six or fewer distinct strings that SOS is to replace with newstring (1 to 6). The total number of characters in all the oldstrings cannot exceed 200.

newstring1...newstring6
: Specifies six or fewer strings that SOS is to substitute for oldstring (1 to 6). The total number of characters in all the newstrings cannot exceed 200.

Options:

,D
: Tells SOS to enter Decide mode.

,N
: Tells SOS to print only the number of the line in which the substitution is made.

,E
: Tells SOS to accept only an exact match (upper- and lowercase letters must agree) in the oldstring(s).

Current line:

After SOS completes a Substitute command, it sets the current line to the last line in which a substitution was made. If SOS does not find the string to be replaced, it does not change the current line.

Common Short Forms:

You must issue a complete Substitute command before using these forms.

S(RET)

Perform substitution(s) on one line: the first line containing the current oldstring. Search from the current line to the end of the current range.

(ESC)
↓
S$.(RET)

Perform substitution(s), using the current oldstring and newstring, on the current line.

(ESC)
↓
S$ (RET)

Perform substitution(s), using the current oldstring and newstring, on the first line containing oldstring. Search from the current line to the end of the file.

This command substitutes an arbitrary string of characters (newstring) for another arbitrary string (oldstring) within a given range of lines, and prints each line in which a substitution is made.

An option (,D) invokes Decide mode, in which SOS displays each line for which a substitution is appropriate -- with the change already made -- and asks you to decide the disposition of the line. Decide mode is described fully in Section 2.8.15.3.

SOS remembers the strings you specify in a Substitute command so that you can reuse them in a later command. For example:

```
          (ESC) (ESC)
            ↓     ↓
*sDoctor$Dr.$/1:/2(RET)
```

This command tells SOS to substitute the string Dr. for all occurrences of Doctor on pages 1 and 2. Then, the following command tells SOS to make the same substitution on page 3.

```
(ESC)
  ↓
*%$/3(RET)
```

If you omit the search strings as shown above before you have issued a complete Substitute command, SOS prints ILLEGAL SEARCH STRING GIVEN.

A Substitute command with no arguments (assuming you have already issued a complete command) performs the substitution in only one line: the next line, starting from the current line, in which oldstring occurs. The search proceeds, if necessary, until the end of the range specified in the last complete Substitute command.

One exception applies: if the last Substitute command operated on the current line, then S(RET) tells SOS to search from the line **after** the current line. Thus, having issued a complete Substitute command, you can perform the same substitution one line at a time throughout the range by repeatedly typing S(RET).

SOS prints a blank line and PAGE n. each time it makes a substitution on a new page.

2.8.15.1 **Multiple Substitutions** - SOS can perform substitutions for several string-pairs at a time. That is, oldstring1 is replaced by newstring1; oldstring2 by newstring2; and so on. Use the second usage form above as follows:

1. Specify all oldstrings first, and follow each oldstring with (RET).
2. SOS prompts with S* to remind you that you are within the Substitute command.
3. After typing the last oldstring, type (ESC) and the first newstring. Follow each newstring with (RET). SOS prompts with S*.
4. After specifying the last newstring, type a final (ESC).
5. Complete the command by typing the range (and options, if desired).

The limit for multiple substitutions is six oldstring-newstring pairs. The total length both of all oldstrings, and (independently) of all newstrings, cannot exceed 200 characters.

For example, suppose you want to substitute the variable X6 for all occurrences of X in a FORTRAN program, and likewise Y6 for Y. X and Y are always space-delimited. Use the following Substitute command:

```
*S(SP)X(SP)(RET)
        (ESC)
        ↓
S*(SP)Y(SP)$(SP)X6(SP)(RET)
           (ESC)
           ↓
S*(SP)Y6(SP)$/1:/*(RET)
```

You can specify fewer newstrings than oldstrings. If you do, SOS substitutes the last newstring for all the extra oldstrings. For example, to substitute the word "sheep" for all occurrences of "goat" and "goats" on page 1, type:

```
*ssoats(RET)
   (ESC)  (ESC)
   ↓      ↓
S*goat$sheep$/1(RET)
```

Note that in the case of two oldstrings, one of which is a leading subset of the other, you must give the longer oldstring first.

2.8.15.2 **Substitute Options** - The Substitute options (,D ,N ,E) can appear in any order at the end of a substitute command. Any options used in a complete Substitute command remain in effect for later S(RET) commands.

SOS **always** distinguishes uppercase from lowercase characters in newstring(s). The ,E option only affects SOS's interpretation of oldstring(s).

For example, suppose you want to change the name of the utility routine in the following line:

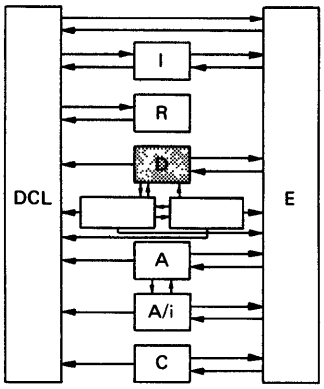
```
00330    The utility routine EXE is executed by
```

The following Substitute command effects the change:

```
                ESC ESC
                 ↓   ↓
*sEXE$ XQT$ 330,e RET
00330   The utility routine XQT is executed by
*
```

The EXACT switch controls this exact pattern-matching feature. If EXACT is on, every Substitute (and Find) command requires an exact pattern match. When SOS is initiated, EXACT is off. Section 2.9.2 describes the EXACT switch.

2.8.15.3 **Decide Mode** - SOS enters Decide mode if you type the option ,D after the range in a Substitute command. (SOS also enters Decide mode if the DECIDE switch is on, that is, if you have issued a /DECIDE command.) In this mode you can decide on a line-by-line basis whether or not to make each substitution.



SOS prints each line in which it makes a substitution, and prompts with D*. Type one of the eight single characters shown in Table 2-5. No RET is needed. If you type any character other than those listed, SOS repeats the prompt.

Table 2-5
Decide-Mode Commands

Form	Meaning
SP	Yes, make the change as shown. SOS continues in Decide mode with the next line to be changed, if any remains in the original range.
DEL	No, do not make the change. The line remains unaltered. SOS continues in Decide mode.
A	Alter: enter Decide Alter mode, with the change already made. Once in Decide Alter mode, you can type any legal Alter-mode commands. When you end Decide Alter mode, SOS continues Decide mode by displaying the next changed line (if any remains in the original range).
E	End: do not make the change, and immediately leave Decide mode (return to Edit mode). No further substitutions are made in the original range.
G	Go: yes, make the change, and make all further changes in the range automatically, without Decide-mode intervention.
N	Same as DEL: no, do not make the change.
Q	Quit, same as E: do not make the change, return immediately to Edit mode.
Y	Same as SP: yes, make the change.

T
Transfer

2.8.16 Transfer (T)

Format:

Tposition,range[,increment1[,increment2]] RET

Arguments:

position
: Specifies the destination for the transferred lines. If the line number specified by position already exists, the transferred lines are inserted after position. If not, the lines are inserted beginning at position.

range
: Specifies the source range of the lines to be transferred.

increment1
: Specifies the line-number increment for SOS to use for the transferred lines. If the source range contains page marks, increment1 is used only up to the first page mark. If you omit increment1, SOS uses the current line-number increment.

increment2
: Specifies the line-number increment for SOS to use for the lines following the final page mark in the source range. If you omit increment2, SOS uses the current line-number increment.

Current line:

After SOS completes a Transfer command, it sets the current line to the last line moved.

The Transfer command moves a block of text from one place in a file to another. When you give a Transfer command, SOS acts as if you had given an Input command at the given location, and treats each text line in the specified range as if you had typed it in. The text is deleted from its original place in the file. Contrast this command with the Copy command, which moves a block of text without deleting it (see Section 2.8.2).

SOS takes steps to make sure that line numbers are in sequence after a Transfer operation. The rest of this section gives details on line numbers before and after a Transfer command. Figure 2-4 shows an example of a Transfer command.

SOS chooses a small enough line-number increment to allow all the lines in the range to fit after the destination position, using the following protocol:

1. SOS checks whether all the lines can fit if it uses the current line-number increment.

2. If the lines will not fit with the current line-number increment, SOS checks whether the lines can fit if it uses the increment you supply (increment1).

3. If the lines will not fit with increment1 (or if you have not supplied increment1), SOS tries to find a smaller increment that can make the lines fit.

SOS cannot choose a small enough increment if the number of lines to be moved is larger than the numerical difference between the line number at position and the number of the next line in the file. In this case, SOS inserts page marks. First, it transfers as many lines as will fit, using either the current increment or your increment1 (if supplied), inserts a page mark, and transfers the rest of the source lines with their line numbers intact. Then SOS inserts a second page mark.

SOS tells you what increment it has used with the message INC1=n. If it inserts a page mark, the message is INC1=ORDER n AND P/M INSERTED.

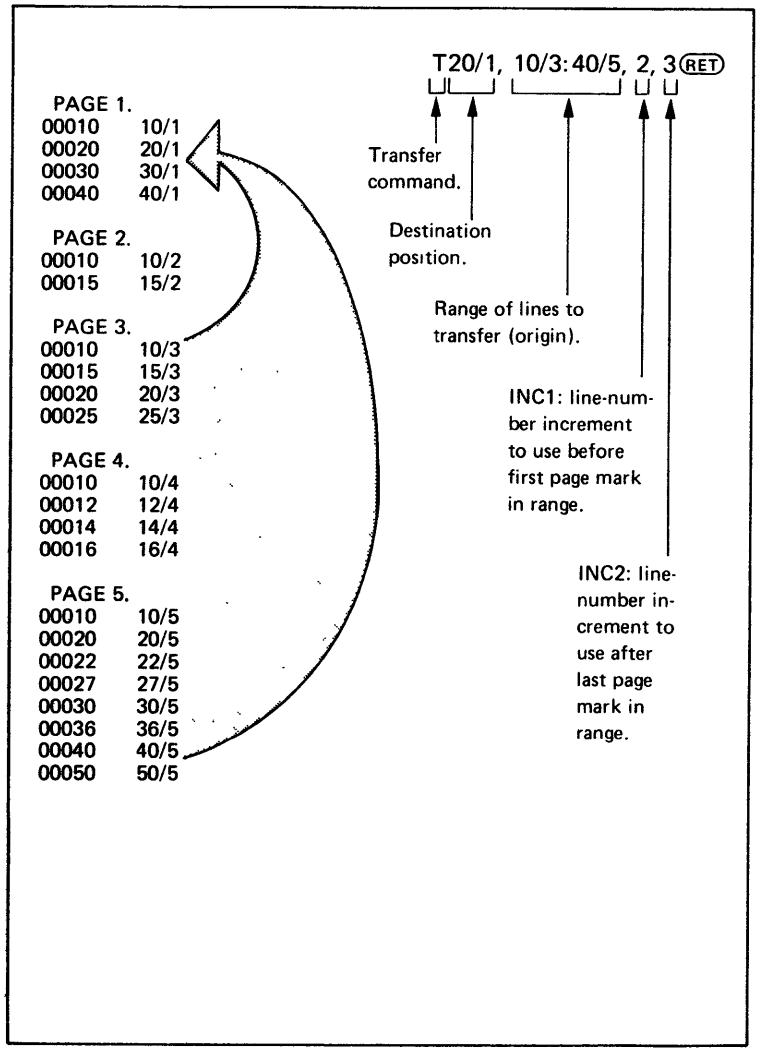
If the range of lines to be transferred already contains page marks, SOS renumbers the transferred lines only up to the first page mark. The page mark is copied, and all the lines up to the last page mark in the range are transferred with their line numbers intact.

For the lines after the last page mark, SOS follows a protocol similar to the one it uses for lines before the first page mark:

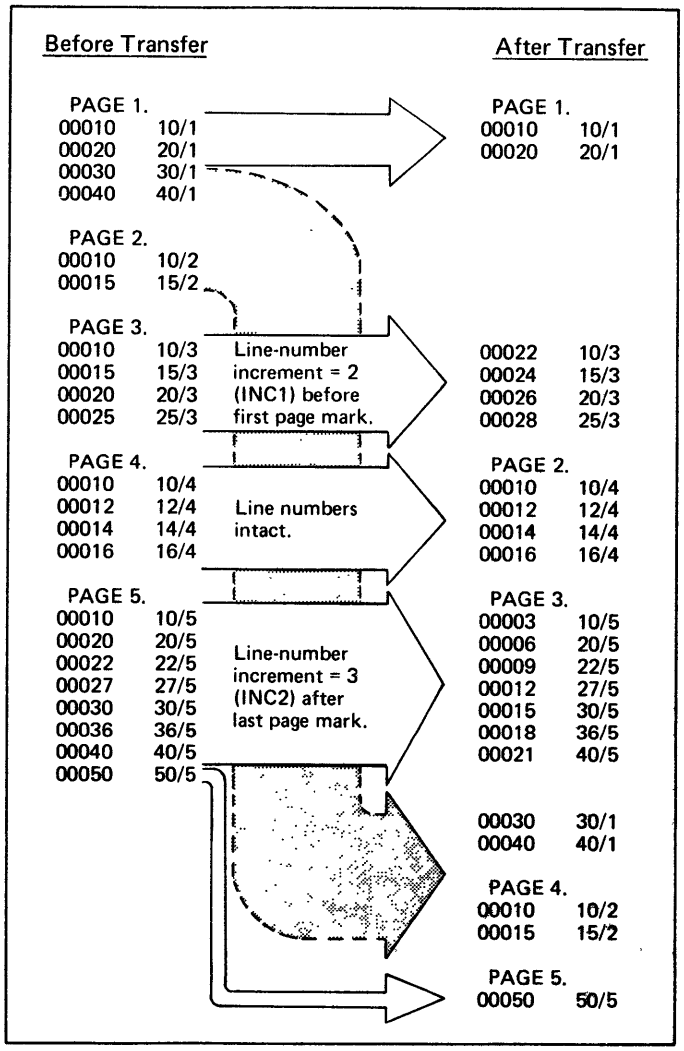
1. SOS moves them using the default line-number increment, if they can fit.

2. If the lines will not fit with the default line number increment, SOS uses the increment you specify (increment2).

3. If the lines will not fit with increment2 (or if you have not supplied increment2), SOS chooses a smaller increment.



The Transfer command is issued to move part of the file to another location (as indicated above by the arrow). The file in this example has as its text, for each line, the corresponding line number and page number.



The above shows the origin and destination of each line in the file before and after the Transfer command, respectively. SOS's line numbering would be the same for an equivalent Copy command, except that a copy of the lines would be left in their original place (before line 50, page 5).

Figure 2-4 Transfer Operation to Move Part of a File to Another Location

If SOS cannot choose a small enough increment, it inserts another page mark, moves the lines with numbers intact, and inserts a final page mark.

If the source range contains page marks, SOS tells you the increment it uses for the lines copied after the final page mark. The message is INC2=n if SOS does not have to insert a final page mark, and INC2=ORDER n AND P/M INSERTED if it does insert one.

If the specification of the source range contains the shorthand symbol ^, meaning the first line on a page, SOS duplicates the page mark that precedes the first line before it begins to move the text. If the range specification contains *, meaning the last line, SOS duplicates the following page mark as well.

Note that SOS duplicates the page mark(s) whether you specify the symbols ^ and/or *, or whether SOS supplies them as defaults. Thus, SOS duplicates the preceding page mark if you type the range

```
/3:50/4
```

This is because the initial line number for page 3 defaults to ^. SOS copies the succeeding page mark if you type the range

```
20/3:/4
```

This is because the final line number for page 4 defaults to *. If you type only a single page number for the range, SOS copies both the preceding and the succeeding page marks for that page. For example:

```
/5
```

This is because SOS expands /5 to ^/5:*/5.

W
Save World

2.8.17 Save World (W)

Formats:

W[:file-spec] RET
WB[:file-spec] RET

Arguments:

file-spec
: If given, specifies the output file. If you omit file-spec, SOS uses the current output file specification.

WB
: Tells SOS not to increment the version number of the output file. The previous version of the output file (or the input file, if you have not used W before) is overwritten.

Current line:

The Save World command does not affect the current line.

The Save World command directs SOS to write the current version of a file being edited to disk. SOS tells you the name of the file it has written. After the command, you remain in Edit mode.

Use the Save World command frequently as a precaution against system malfunction or against your own errors. If a catastrophe happens, you only lose the work done since the last time you saved "everything in the world." Section 2.7 gives an overview of "defensive editing."

Each time you use the W command, the current contents of the file are overwritten on the output file. Thus, if you repeatedly use the W command without giving a different file specification, you are repeatedly updating a single back-up file. To obtain a chronological string of back-up files, use W with a different file specification each time.

WB tells SOS not to increment the version number before writing the output file. This option of the Save World command is effective only if it is the first W command given. That is, once a W command has been issued, the version number of the output file has been incremented. All subsequent W or WB commands use that same version number. Only if WB is the first Save World command will it overwrite the original input file; otherwise WB is eguivalent to W. SOS overwrites the input file with the current contents of the file you are editing, and returns you to Edit mode.

X
eXtend

2.8.18 eXtend (X)

Format:

X[range][,N]RET

Argument:

range
Specifies the range of lines to be extended one at a time. If you omit range, the current line is extended.

Option:

,N
Tells SOS to print only the line number of each line in the range.

Current line:

After SOS completes an eXtend command, it sets the current line to the last line in the range that was extended.

The eXtend command adds text to the end of lines. It does this by acting as if you had typed A (to enter Alter mode), TAB (to reach the end of the line), and I (to begin inserting text). Alter mode is described in detail with the Alter command in Section 2.8.1.

SOS enters Alter/insert mode for each line in the range that you specify. It prints the line and waits in Alter/insert mode. If you specify ,N, SOS prints only the line number. At any time you can type ESC to end insertion and then give any Alter-mode command. For example, typing L moves the intraline pointer to the beginning of the line, so you can make changes before the text you have just inserted.

If you type DEL while inserting text, the previous character is deleted. DEL echoes as a pair of backslashes followed by the deleted character. Successive DELs delete and echo previous characters in turn. The next non DEL character that you type is echoed as a second pair of backslashes followed by the character. If you type more DELs than the number of characters you have inserted after the eXtend command, SOS continues deleting characters to the left, back before the end of the original line.

If you insert so much text that the overall length of the line exceeds 500 characters, SOS prints the message LINE IS TOO LONG and returns you to Edit mode. Any changes that you may have made to the line up to that point are discarded.

If you type (LF) after the eXtend command (that is, while in Alter/insert mode), SOS creates a new line to receive the rest of the inserted text. The number of this new line is determined as follows:

1. The current line-number increment is added to the current line number.

2. If an existing line intervenes between the current line number and this new number, the new line is numbered halfway in between.

3. If the next existing line is numbered higher by 1 than the current line, a bell sounds and no new line is created.

If no errors occur and the new line is created, the intraline pointer is positioned at the beginning of the new line; SOS is still in Alter/insert mode.

After an eXtend command, while in Alter/insert mode, any character you type (with seven exceptions) is inserted into the text line. Many control characters do not have their usual effect when entered in Alter/insert mode. For example, (CTRL/S) and (CTRL/Q) have no effect on terminal output. They are simply inserted into the text line without interpretation.

The following seven characters are not treated as text in Alter/insert mode:

(ESC)	ends insertion.
(RET)	ends the eXtend command for this line.
(DEL)	deletes the previous character.
(LF)	creates a new text line.
(CTRL/C)	returns SOS to edit mode.
(CTRL/R)	retypes the line.
(CTRL/Y)	gets the attention of DCL.

. (period)
Move Position

2.8.19 Move Position (.)

Format:

.position RET

Argument:

position
Specifies the new current position (line/page).

Current line:

After SOS completes a Move Position command, it sets the current line to the line specified by the position argument.

This command resets the current line. If you specify a nonexistent line or page number, SOS prints NO SUCH LINE EXISTS, and does not change the current line. Otherwise, the only response is the Edit-mode prompt *.

Examples:

*.220/2 RET

Set the new current position to line 220, page 2.

*..-5 RET

Set the new current position to the fifth line before the present position. Note that, if at present the current line is less than five lines from the beginning of a page, then ^/. is the current position after the command.

*../.+2 RET

Set the new current position to the line that has the same number as the current line, on the page that is two pages beyond the current page.

= (equal) Give Parameter

2.8.20 Give Parameter (=)

Format:

=parameter RET

Argument:

parameter
Specifies any legal SOS parameter or switch. Parameters and switches are listed in Tables 2-6 and 2-7 (see Section 2.9).

Current line:

The Give Parameter command does not affect the current line.

This command returns the value of SOS parameters and switches. SOS **parameters** can have values of integers, strings, characters, or positions (line/page). A **switch** is an SOS internal variable that can have one of two values: on or off.

Tables 2-6 and 2-7 in Section 2.9 list and define all legal SOS parameters and switches. The following examples show some useful ones.

Examples:

```
*=bis RET
4
*
```

Print the page number of the largest page in the file.

```
*=NAME RET
DBB2:[MANN]PAYROLL.MST;3
*
```

Print the current specification of the output file. This is the filespec used by the Save World and End commands.

```
*=STRING RET
  FIND-
isomer
  SUBSTITUTE-
stochastic
  FOR-
random
*
```

Print the values of the current Find and Substitute strings. These are handy to know when using these commands repeatedly through a file. The string identified as SUBSTITUTE- is newstring, and FOR- is oldstring (see the description of the Substitute command in Section 2.8.15).

/ (slash)
Set Parameter

2.8.21 Set Parameter (/)

Format:

/parameter[:value] RET

Arguments:

parameter
Specifies any SOS switch or parameter that can be set (see Tables 2-6 and 2-7 in Section 2.9).

value
Specifies a legal value for that parameter. See the descriptions of the individual parameters in Section 2.9.1.

Current line:

The Set Parameter command does not affect the current line.

This command sets the value of SOS parameters and switches. You are allowed to set any of the switches, but you can set only nine of the 16 parameters. Table 2-6 in Section 2.9 shows which parameters you can set.

After setting the requested parameter or switch, SOS prints the Edit-mode prompt *.

The examples below demonstrate setting some of the more frequently used SOS parameters and switches. All are described in Section 2.9.

Examples:

*/isave:10 RET

Perform an automatic Save World command after every 10 lines added in Input mode.

*/expert RET

The user claims to be an expert at using the editor: issue shorter error messages; do not request confirmation before deleting text; and do not display deleted text in Alter mode.

*/PLINES:23 RET

Print 23 lines of text, starting at the current line, when SOS receives P RET. This is a useful value of PLINES for users with CRT terminals that display 24 lines of text on the screen (especially if the terminal's speed has been set to a high baud rate, for instance 1200 baud or above).

@ (at) Command File

2.8.22 Command File (@)

Format:

@file-spec(RET)

Argument:

file-spec
Specifies a file containing SOS commands.

Current line:

After SOS completes a Command File command, it sets the current line to the last line modified by a command in the file.

Command File lets you place a repetitive or frequently used series of commands in a file. SOS executes the commands from the file as if they came from the keyboard.

Once SOS begins executing a Command File command, it performs all the commands in the file, returning control to the keyboard only when it reaches the end. However, SOS prints the following items on the terminal:

- Error messages
- Printouts from the Print command
- Lines found by the Find command
- Changes made by the Substitute command
- Lines altered by the Alter command

SOS also requests input from the terminal if the command file contains a Substitute command that places the editor in Decide mode.

If the file type field of the command-file specification is CMD, you can omit the file type when giving the file specification.

Command files can be nested to a depth of three levels. That is, a command file can contain a Command File command; that file can contain another Command File command. If you attempt to nest command files to a depth of four or more, SOS prints MAXIMUM COMMAND FILE DEPTH EXCEEDED and issues the Edit-mode prompt *.

You can use command files to shorten the typing of common commands. For example, you might create two files named L.CMD and W.CMD as shorthand for commonly used Print commands:

```
L.CMD:
  P*-5/*!6

W.CMD:
  P,-3!6
```

If these two files are in your directory, you can type @L(RET) to see the last six lines in a file, and type @W(RET) to see a "window" of six lines surrounding the current line.

Another use for Command File is in repetitive editing. Consider the task of deleting the final three characters on each of 400 lines on a page. This task would be tedious to perform by hand, and prone to error as well. The example below shows how to build and use a command file to do the required deletion.

Example:

In this example, the user (whose name is Lyles) generates a command file named DEL.CMD. The first line contains an X (eXtend) command with a range of 100 lines. (Lyles gives four Command File commands to delete the 400 lines.) The remaining 100 lines in DEL.CMD contain identical Alter-mode command sequences that perform the deletion on each line in turn.

Lyles uses the eXtend command, instead of the Alter command, to suppress typeout of the lines being altered. The ,N option of eXtend allows this. Since eXtend leaves SOS in Alter/insert mode, the first Alter-mode command is (ESC), which returns SOS to Alter-mode command level. The subsequent command deletes the previous three characters (the last three characters in the line).

To get an escape character into the file, Lyles uses the Set Parameter command /ESCAPE to set the logical escape character to the plus sign (+). Thus, when Lyles inserts line 200 and types +, SOS inserts an escape character into the file. (For details, see Section 2.9.1.)

After entering the line of Alter-mode commands, Lyles uses the Copy command to replicate the line the required 100 times. Note that, after seven copy commands, there are 128 replicated lines; Lyles deletes the excess.

```
$ EDIT DEL.CMD<RET>

INPUT:DBA2:[LYLES]DEL.CMD;1

                    <ESC>
00100    X.FLOG.N$
*.S/GET 14<RET>
*;*<RET>
00200    1-3D<RET>
       <ESC>
00300    $
*C*,200<RET>
INC1=100
*C*,200;*<RET>
INC1=100
*C*,200;*<RET>
INC1=100
*C*,200;*<RET>
INC1=100
*C*,200;*<RET>
INC1=100
*F*<RET>
03300    -3D
*C*,200;*<RET>
INC1=100
*C*,200;*<RET>
INC1=100
*F*<RET>
12900    -3D
*D.0200;*<RET>
28 LINE(S) DELETED (10200/1:12900).
*E<RET>

[DBA2:[LYLES]DEL.CMD;1]

$
```

Now Lyles edits the file in which the deletions are needed. The 400 lines are on page 3, numbered 10 to 4010. Because the eXtend command in DEL.CMD uses the current line (.), Lyles sets the current line to the next unedited line before typing @DEL. Immediately after giving each Command File command, Lyles types <CTRL/O> to suppress all printout. SOS completes the command and issues the Edit-mode prompt.

```
$ EDIT DIREC.COR<RET>

EDIT:DBA2:[LYLES]DIREC.COR;12
*.,/3<RET>
*@DEL<RET><CTRL/O>
*P.,!2<RET>
01010    Gunther, Jay .........
01020    Gurska, Andrew ....... 22
*@DEL<RET><CTRL/O>
*P.,!2<RET>
02020    Mosberg, Tom .........
02030    Modeen, Alice ........ 41
*@DEL<RET><CTRL/O>
*.,.+1<RET>
*@DEL<RET><CTRL/O>
*P.-1:*<RET>
04000    Zosel, Marie .........
04010    Zysk, Raymond ........
*
```

2.9 SOS PARAMETERS AND SWITCHES

SOS maintains a number of internal parameters and switches that control different aspects of its operation. You can set and reset some of the parameters and switches from the terminal. Tables 2-6 and 2-7 list the parameters and switches.

A **switch** in SOS is an internal variable that can have one of two values: on or off. Switches are set with the Set Parameter command (/) and interrogated with the Give Parameter command (=). All switches can be turned off by prefixing NO to their names in a Set Parameter command.

Figure 2-5 shows the SOS switches and their initial values; minimum abbreviations are given in uppercase letters. A **parameter** can have a range of values. Most parameters have integer values:

BIG	INCREMENT	ISAVE
LENGTH	PLINES	SAVE
START	STEP	SUBSTITUTE

Four parameters have values that are text strings:

ERROR
ID
NAME
STRING

Two parameters have values that are positions (line/page pairs):

.
LOCATION

Two parameters have values that are single characters:

ESCAPE
MATCH

Some of the parameters and switches can be specified as qualifiers in the initial DCL EDIT command. They can also be specified by using the symbol-assignment mechanism (see Section 2.9.3). The parameters and switches that you can preset in the EDIT command are:

INCREMENT	STEP
ISAVE	DECIDE
PLINES	EXPERT
SAVE	LINE
START	LOWER
	READONLY

Table 2-6
SOS Parameters

Name[1]	Set from Edit Mode?	Preset via DCL?[2]	Initial Value	Meaning
BIG	n		–	Displays highest page number
ERROr	n		(null)	Displays last error message
ESCApe	y		(null)	Sets character for input/output of escapes
ID	n		–	Displays current SOS version number
INCRement	y	y	100	Sets increment for numbering inserted lines
ISAVe	y	y	0	Controls auto-W on inserts
LENGth	y		55	Sets page size for List command
LOCAtion	n		^/1	Displays first line in edit buffer
MATCh	y		(null)	Sets special pattern-matching flag character
NAME	n		–	Displays output file specification
PLINes	y	y	16	Sets number of lines printed by P RET
SAVE	y	y	0	Controls auto-W on commands
STARt	y	y	100	Sets starting line number
STEP	y	y	100	Sets increment for initial line numbering
STRIng	n		(null)	Displays current Find and Substitute strings
SUBStitute	n		0	Displays number of matches in last Substitute
.	n		00000/1	Displays current position

[1] Minimum abbreviation is given in uppercase letters.
[2] You can specify these parameters as qualifiers in the initial DCL EDIT command, or by using the symbol-assignment mechanism (see Section 2.9.3).

Table 2-7
SOS Switches

Name[1]	Initial Value	Preset via DCL?[2]	Meaning
BAK	on		Tells SOS to create back-up file
DECIde	off	y	Sets auto Decide mode on Substitute
EXACt	off		Requires exact case match for Find, Substitute, content-specification
EXPErt	off	y	Declares experienced SOS user
LINE	on	y	Tells SOS to use existing line numbers
LOWEr	on	y	Tells SOS to accept uppercase, lowercase as is
READonly	off	y	Starts SOS in Read-only mode
SEPArator	off		Tells SOS to treat _$. as alphanumeric
SEQUence	on		Leaves output-file line numbers

[1] Minimum abbreviation is given in uppercase letters.
[2] You can specify these switches as qualifiers in the initial DCL EDIT command, or by using the symbol-assignment mechanism (see Section 2.9.3). Turn off any switch by using the letters NO before its name; thus, NOLINE turns off the LINE switch.

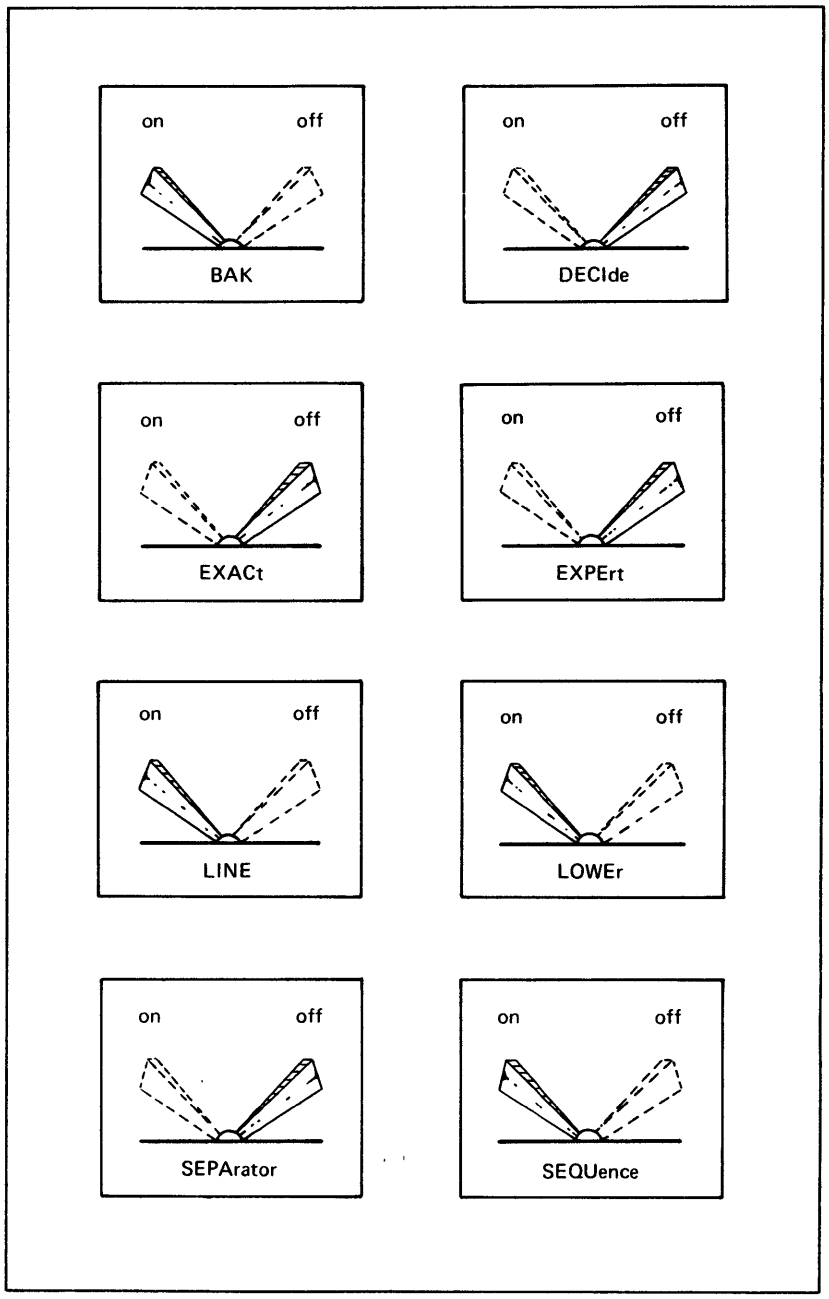


Figure 2-5 SOS Switches and Their Initial Values

2.9.1 Parameters

BIG

Value:
Integer: the largest page number now in the file.

Default:
Null.

Set:
Cannot be set by user. SOS resets BIG when you add or delete pages, and (if necessary) when you interrogate it with =BIG<CR>.

Interrogate:
Use =BIG (RET)

When you give the command =BIG for the first time, SOS searches through the entire file, inserting page marks if it finds lines out of order, until it reaches the end of the file. The following output is typical.

```
* =BIG (RET)

LINES OUT OF ORDER
P/M INSERTED FOR PAGE 2

LINES OUT OF ORDER
P/M INSERTED FOR PAGE 5
7
*
```

In this example, SOS returns the value 7 as the highest numbered page in the file. The message LINES OUT OF ORDER might result from concatenating line-numbered files, for example with the DCL command COPY.

ERROR

Value:
String: the last error message SOS printed.

Default:
Null.

Set:
Cannot be set by user. SOS resets ERROR each time it prints an error message.

Interrogate:
Use =ERROR (RET)

SOS always prints the long form (that is, the NOEXPERT form) of the last error message. It does this even if the EXPERT switch is on.

ESCAPE

Value:
Character: when input, causes SOS to insert an escape character into the file; also, the character that SOS prints when it encounters an escape character in text.

Default:
Initially null. When you do not specify a character (/ESCAPE RET), the default is $ (dollar sign).

Set:
Use /ESCAPE:x RET

Interrogate:
Use =ESCAPE RET

This parameter defines a "logical escape" character that lets you insert an escape character into a text file, and lets SOS display escape characters in text that it types out. One reason for inserting escape characters into text files is to use the Find and Substitute commands in command files (see Section 2.8.22).

This parameter is initially null, so you cannot insert escape characters into the text. When you issue the following Set Parameter command, SOS uses the default value of $ as the logical escape character.

```
*/ESCAPE RET
```

The logical escape is used in three ways:

- SOS inserts an escape character into the text whenever you type $ in Input mode, in Alter/insert mode, or in the newstring of a Substitute command.
- SOS prints $ whenever it encounters an escape character in text to be typed out.
- SOS searches for an escape character when you type $ in a Find command or in the oldstring of a Substitute command.

You can select another character to act as the logical escape character by typing:

```
*/ESCAPE:x RET
```

where x represents the new logical escape character.

NOTE

You cannot use any of the following characters as a logical escape:

; : / [ ] =

These characters are meaningful in file specifications.

To disable the logical escape feature, type:

```
*/ESCAPE: RET
```

Note that the colon must be typed.

ID

Value:
String: the current version number of SOS.

Default:
The current version.

Set:
Cannot be set by user.

Interrogate:
Use =ID(RET)

SOS responds to the command =ID(RET) with its current version number. If you experience problems with the text editor software, ascertain the version number of SOS so you can report it to DIGITAL service personnel. For example:

```
*=ID(RET)
SOS  V02.04A
*
```

INCREMENT

Value:
Integer: the line-number increment SOS uses to assign the next line number for input text.

Default:
100

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /INCREMENT:n<CR>. SOS changes INCREMENT when you give an Input or Replace command with a second argument (comma-delimited) to update the line-number increment.

Interrogate:
Use =INCREMENT(RET)

INCREMENT is the line-number increment SOS uses to calculate line numbers for new lines created in Input or Alter/insert mode. You can set this parameter directly by using the Set Parameter command. However, the more common way to change its value is by supplying an increment argument to the Input or Replace commands.

ISAVE

Value:
Integer: number of lines of input text between automatic file saves.

Default:
0 (auto-save turned off).

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /ISAVE:n(RET).

Interrogate:
Use =ISAVE RET

This parameter controls one of SOS's automatic file-saving options (the other is controlled by SAVE). When you set ISAVE -- for example, /ISAVE:10 -- SOS automatically performs a W command each time you insert 10 more lines into the file using the Insert or Replace commands.

ISAVE initially has the value 0, so auto-saving is disabled on inserts. To disable ISAVE once you have set it, reset it to 0.

LENGTH

Value:
Integer: page length for the List command.

Default:
55

Set:
Use /LENGTH:n RET

Interrogate:
Use =LENGTH RET

LENGTH represents the page length for printer listings generated by the List command. Its initial value is 55 lines. Thus, SOS starts a new page in the printer listing whenever it encounters a page mark, and also whenever it has printed 53 lines without encountering a page mark (the listing heading requires two lines).

LOCATION

Value:
Line/page: the position of the beginning of SOS's edit buffer.

Default:
^/1

Set:
Cannot be set by user. SOS keeps track of LOCATION as it reads from the input file and writes to its temporary files.

Interrogate:
Use =LOCATION RET

The edit buffer contains all the text that is immediately accessible to SOS. If you issue a command that affects lines outside this buffer, SOS must read text from the input file before performing the command. See Section 2.13 on SOS Internals.

For example, if you request the printout of a line earlier in the file than the position LOCATION, SOS must write the remainder of the file (after the current position) to its temporary file, and then read from the input file up to the line you requested.

MATCH

Value:
Character: the flag character for special pattern-matching in Find and Substitute commands.

Default:
Initially null. When you do not specify a character (/MATCH(RET)), the default is ? (question mark).

Set:
Use /MATCH[:x](RET)

Interrogate:
Use =MATCH(RET)

This parameter defines a flag character for SOS's special pattern-matching feature (see Section 2.10). MATCH is initially null, so special pattern-matching is disabled.

When you issue the following Set Parameter command, SOS uses the default value of ? as the flag character.

*/MATCH(RET)

In Find and Substitute strings, certain groups of characters beginning with ? are interpreted in a special way (see Section 2.10).

You can select another character to act as the special pattern-matching flag character by typing:

*/MATCH:x(RET)

where x represents the new flag character.

To disable special pattern-matching, type:

*/MATCH:(RET)

Note that the colon must be typed.

NAME

Value:
String: the specification of the output file.

Default:
The specification of the input file.

Set:
Cannot be set by user. SOS changes NAME when you issue a Save World command, giving a new file specification.

Interrogate:
Use =NAME(RET)

After you issue a Save World command, the parameter NAME contains the file specification that SOS would use for the output file if you gave an End command (without naming a new file). Before you have issued a Save World command, NAME contains the input file specification. However, if you explicitly gave an output file specification when you initiated SOS (/OUTPUT:file-spec), NAME initially contains this specification.

PLINES

Value:
Integer: the number of lines SOS prints upon receiving a Print command without arguments.

Default:
16

Set:
Append as a qualifier to the EDIT command; or set by the symbol assignment mechanism; or use /PLINES:n(RET).

Interrogate:
Use =PLINES(RET)

SAVE

Value:
Integer: number of text-changing commands between automatic file saves.

Default:
0 (auto-save turned off).

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /SAVE:n(RET).

Interrogate:
Use =SAVE(RET)

This parameter controls one of SOS's two automatic file saving options (the other is controlled by ISAVE). When you issue the command /SAVE:20, for example, SOS automatically performs a Save World command each time you issue the 20th text-changing command (listed below).

Commands that do not change the file are not included in this count. Text-changing commands are defined as the commands that are illegal in Read-only mode:

Alter	Copy	Delete
Insert	Join	Kill Page Mark
Mark	reNumber	Replace
Substitute	Transfer	eXtend

START

Value:
Integer: the line number SOS gives to the first line in a file, and/or to the first line on a page, in an unnumbered file.

Default:
100

Set:
To override SOS's initial default value, append as a qualifier to the EDIT command, or set by the symbol-assignment mechanism; or use /START:n(RET).

Interrogate:
Use =START(RET)

If you reset START midway through editing a file that was initially unnumbered but that contained page marks, SOS numbers subsequent pages (that it has not yet encountered) using the new starting line number.

STEP

Value:
Integer: the line-number increment SOS uses to assign line numbers to an unnumbered file.

Default:
100

Set:
To override SOS's initial default value, append as a qualifier to the EDIT command, or set by the symbol-assignment mechanism (see Section 2.9.3); or use /STEP:n(RET).

Interrogate:
Use =STEP(RET)

STEP is the default line-number increment SOS uses to number the lines of an unnumbered input file. SOS assigns line numbers as it reads in the lines. Therefore, you can change STEP at any time to affect the line-number increment for the lines that SOS has not yet accessed in an unnumbered file.

STRING

Value:
String: the current Find and Substitute strings.

Default:
Null.

Set:
Cannot be set by user. SOS updates STRING whenever you issue a new Find or Substitute command.

Interrogate:
Use =STRING(RET)

The parameter STRING contains the values of three search strings: one for the Find command and two for the Substitute command. Any of these search strings can contain multiple strings, if such strings were specified in the last Find or Substitute. For example:

```
*=STRING(RET)
  FIND-
Alter mode          Find command strings
Alter-mode
  SUBSTITUTE-
Decide mode         Substitute command newstrings
Decide-mode
  FOR-
decide mode         Substitute command oldstrings
decide-mode
*
```

It is sometimes useful to issue =STRING(RET) when you are using the Find and/or Substitute commands without arguments; that is, when you are searching for strings specified in a previous complete command.

SUBSTITUTE

Value:
Integer: the number of substitutions performed by the last Substitute command.

Default:
0

Set:
Cannot be set by user. SOS updates SUBSTITUTE after every Substitute command, whether successful or not.

Interrogate:
Use =SUBSTITUTE(RET)

This parameter is useful because it tells you how many substitutions SOS performed in a Substitute command with the ,N option (in which the lines changed are not printed out).

If you entered Decide mode with the last substitute command, and if you entered Decide Alter mode for any line, the SUBSTITUTE parameter reflects the total number of possible substitutions for that line. This is true even if, in Decide Alter mode, you undo one or more of the substitutions.

. (Period)

Value:
Line/page: SOS's current position.

Default:
00000/1

Set:
Use .position(RET). SOS updates this parameter after every command that affects a line.

Interrogate:
Use =.(RET)

2.9.2 Switches

SOS switches are summarized in Table 2-7 and Figure 2-5. All switches can be turned on with the Set Parameter command; for example:

```
*/Bak(RET)
```

Several of the switches can be specified as qualifiers to the initial EDIT command, or by using the symbol-assignment mechanism (see Section 2.9.3). All switches can be interrogated with the Give Parameter command:

```
*/Bak(RET)
```

You can abbreviate switch names to four characters. In all the examples below, the full switch names are used.

Any switch can be turned off by prefixing NO to its name in the Set Parameter command. For example, to turn off the Decide-mode switch, type:

```
* /NODECIDE (RET)
```

For compatibility with other implementations of SOS, you can use the minus sign (-) interchangeably with NO to turn off a switch. However, you cannot use the minus sign on the initial DCL EDIT command line. It is best to use NO in all cases.

BAK

Meaning:
Determines whether SOS increments the version number of the output file.

Default:
BAK is on.

Set:
Use /BAK(RET) or /NOBAK(RET) (after initialization, SOS does not set this switch).

This switch determines whether SOS increments the version number of the output file when you issue the first Save World command in an editing session, or when you issue End without a previous Save World. If BAK is on, SOS increments the version number. If BAK is off, SOS does not increment the version number, but instead overwrites the input file.

The setting of the switch is not meaningful if you give a new file specification with the Save World or End command. This is because the new file specification obviates the question of whether or not you want a back-up file. The setting is also not meaningful after you issue the first Save World command.

DECIDE

Meaning:
Determines whether SOS automatically enters Decide mode for each Substitute command.

Default:
DECIDE is off.

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /DECIDE(RET) or /NODECIDE(RET) (after initialization, SOS does not set this switch).

If DECIDE is on, SOS enters Decide mode each time you issue a Substitute command. The effect is as if you issued each Substitute command with the ,D option.

The DECIDE switch provides an automatic way of protecting against operator errors. Therefore, a user who is just beginning to learn SOS may want to set this switch.

EXACT

Meaning:
Determines whether SOS requires an exact match (uppercase and lowercase letters) in pattern searches.

Default:
EXACT is off.

Set:
Use /EXACT(RET) or /NOEXACT(RET) (after initialization, SOS does not set this switch).

If EXACT is on, SOS requires an exact pattern match for:

- All Edit-mode Find and Substitute command searches
- All Alter-mode S and K command searches
- All content-specified searches (see Section 2.11)

EXPERT

Meaning:
If EXPERT is on, SOS assumes you are an experienced SOS user.

Default:
EXPERT is off.

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /EXPERT(RET) or /NOEXPERT(RET) (after initialization, SOS does not set this switch).

Declaring yourself an expert SOS user (/EXPERT(RET)) has four consequences:

- The error and informational messages SOS types are more brief than those for nonexperts.
- SOS does not request confirmation before deleting text across page boundaries, or before deleting all text on a page.
- SOS does not report how many lines it deletes when you issue a Delete or Replace command.
- In Alter mode, SOS does not display deleted characters between pairs of backslashes (\\...\\). Thus, the appearance of lines as you edit them in Alter mode corresponds more closely to the actual contents of the lines.

LINE

Meaning:
Determines whether or not SOS uses line numbers that may be present in an input file.

Default:
LINE is on (that is, SOS uses existing line numbers).

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /LINE(RET) or /NOLINE(RET) (after initialization, SOS does not set this switch).

If you turn off the LINE switch by the logical name mechanism, or if you initiate SOS with the qualifier /NOLINE, SOS ignores line numbers that may be present in the file it reads. The file is renumbered. SOS does not print the usual message indicating the absence of line numbers (INPUT ASSUMED UNSEQUENCED).

LOWER

Meaning:
Determines whether or not SOS translates lowercase characters from the terminal to uppercase.

Default:
LOWER is on (that is, lowercase letters are not translated to uppercase).

Set:
Append as a qualifier to the EDIT command; or set by the symbol-assignment mechanism; or use /LOWER(RET) or /NOLOWER(RET) (after initialization, SOS does not set this switch).

A side effect of turning off LOWER is that if the exact character match options (,E) of the Find and Substitute commands are used, it is impossible to match any lowercase character that is already in the file. The EXACT switch, if turned on, has the same side effect.

SEPARATOR

Meaning:
Determines whether SOS treats the three characters ., $, _ as word delimiters if special pattern-matching is enabled for Find and Substitute commands.

Default:
SEPARATOR is off.

Set:
Use /SEPARATOR(RET) or /NOSEPARATOR(RET) (after initialization, SOS does not set this switch).

By setting the switch SEPARATOR, you can make SOS treat the characters ., $, and _ as word-delimiters for purposes of special pattern-matching (see Section 2.10). This feature could be useful in editing a MACRO program, for instance, in which decimal points, dollar signs, and underscore characters appear frequently as parts of symbols.

SEQUENCE

Meaning:
Determines whether SOS leaves line numbers in the output file.

Default:
SEQUENCE is on.

Set:
Use /SEQUENCE(RET) or /NOSEQUENCE(RET) (after initialization, SOS does not set this switch).

If you turn off the SEQUENCE switch and exit with the E command, the effect is as if you exited with ES.

2.9.3 Setting Default Conditions Using Symbol Assignment

You can change the parameter and switch values that apply when SOS is initiated by using the DCL symbol-assignment mechanism. You can issue a symbol-assignment command from the terminal to override the initial defaults temporarily; or you can include the command in your LOGIN.COM file to override the defaults permanently.

SOS's initial default values for parameters and switches are listed in Tables 2-6 and 2-7, and discussed briefly in Section 2.5. You can override these defaults in three ways upon starting the editor.

1. To override the defaults for the current editing session only, use qualifiers to the EDIT command line. For example:

```
$ EDIT/INCREMENT:2 filespec(RET)
```

2. To override the defaults for all editing sessions within the current job (that is, until you log off the system), issue a symbol-assignment command from the terminal:

```
$ SOS:="EDIT/SOS/INCREMENT:2"(RET)
```

3. To override the defaults for all future editing sessions, place a symbol-assignment command (similar to item 2 above) in your LOGIN.COM file.

The symbol-assignment command associates a character string with a symbol. In item 2 above, the character string EDIT/SOS/INCREMENT:2 is associated with the symbol SOS. Typing SOS(RET) or SOS file-spec(RET) then becomes your own private way of initiating the editor with the qualifier /INCREMENT:2. Such a symbol assignment lasts until you log off the system.

To define a symbol that lasts from one job to the next, perform the symbol assignment in the file LOGIN.COM. This command procedure file contains commands that are automatically executed each time you log in (see the VAX/VMS Command Language User's Guide).

You can include more than one SOS parameter or switch in a symbol assignment. Use a minus sign (-) in the last character position to continue a long line.

As an example, suppose you have the following line in your LOGIN.COM file:

```
SOS:="EDIT/SOS/LOWER/STEP:10/START:10/INCREMENT:2/PLINES:10"
```

When you type SOS or SOS file-spec to initiate the editor, the parameters and switches specified in the symbol assignment are applied as qualifiers. Therefore, the following conditions hold:

- Lowercase input is accepted as is, and is not converted to uppercase.
- If SOS reads an unnumbered file, it numbers the lines by 10s, starting at 10.
- SOS uses a line-number increment of 2 for inserted lines.
- SOS prints 10 lines, starting at the current line, upon receiving P(RET).

2.10 SPECIAL PATTERN-MATCHING CONSTRUCTS

One SOS feature that allows great flexibility in searching for strings using the Find and Substitute commands is special pattern-matching. This feature is initially disabled. You control it by means of the MATCH parameter (see Section 2.9.1).

The Set Parameter command /MATCH:x(RET) enables special pattern-matching. The character x acts as a flag to mark the special pattern-matching constructs. A construct is a group of two or more characters beginning with the flag character. When SOS encounters such a construct in a Find or Substitute command, it treats it as a single character with a particular meaning for its pattern search, as detailed below.

If you issue the command /MATCH(RET) without specifying a flag character, SOS uses the question mark (?) by default as the special-constructs flag. The examples given later in this section use ? as the flag.

To disable special pattern-matching, and to divest the flag character of any special meaning, type /MATCH:(RET). Note that you must type the colon.

Table 2-8 lists the special constructs. Some of the constructs are used in **find** strings (the string argument of Find and the oldstring argument of Substitute), while others are used in **substitute** strings (the newstring argument of Substitute). Sections 2.10.1 and 2.10.2 describe these two classes of special pattern-matching constructs. Section 2.10.3 describes combining find-string constructs.

Table 2-8
Special Pattern-Matching Constructs

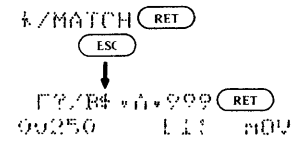
Con-struct	Internal represen-tation	Meaning
Find-string constructs:		
?/	CTRL/T	Match any character
?:	|	Match any separator
?<	CTRL	Match a space or tab
?%x	CTRL/E	Match any character except x
?)x	CTRL/N	Match 0 or more of the character x
?1x	CTRL/V	Match 1 or more of the character x
?9	CTRL/X	Match any alphanumeric character
?!	CTRL/A	Match any letter (A-Z, a-z)
?&	CTRL/F	Match any uppercase letter (A-Z)
?2	CTRL/W	Match any lowercase letter (a-z)
?+	CTRL/P	Match any decimal digit (0-9)
?>	CTRL/]	Match beginning or end of line
?7c	CTRL/^	Match internal representation of c
Substitute-string constructs:		
?"	CTRL/B	Substitute next string matched
?*n?*	CTRL/O	Substitute nth string matched

2.10.1 Find-String Constructs

You can use the constructs given below in the Find command, or in the oldstring of the Substitute command.

?/ Match Any Character

This construct matches any character. As an example of its use, suppose you want to locate variables that start with B in a MACRO program, but you do not want to see lines with labels that start with B. Since labels start in column 1, they are never preceded by any other character. Thus, the following Find command locates the desired variables:

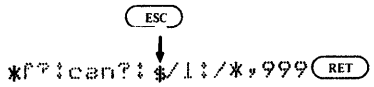


The ,A in the Find command tells SOS to enter Alter mode when a match is found. In Alter mode, the intraline pointer points before the first character of the pattern matched.

?: Match Any Separator

This construct matches any separator, that is, any character except a letter, a number, or the characters ., $, and _. (If the switch SEPARATOR is turned on, these three characters also are considered separators. See Section 2.9.2.) The ?: construct is useful in searching for words that, besides appearing in isolation, may be imbedded in other words as well. For example, suppose you want to

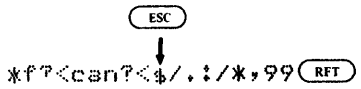
find all occurrences of the word "can"; however, you are not interested in words like incandescent, canonical, and so forth. You can issue the following Find command:



The ?: constructs before and after the word are matched by a tab or a space. Note that, unless the command /SEPARATOR(RET) was given, this Find command does not match the word "can" if it ends a sentence. SOS does not treat the final period as a separator. The Find command does work if the word "can" begins or ends a line, however, because SOS considers the beginning and end of a line to be separators.

?< Match a Space or Tab

This construct matches a space or a tab character. Thus, it represents a subset of the ?: (match any separator) construct described above. The following command has the same effect as the example command in the previous section:



This Find command does not match the word "can" if it begins or ends a line, or if it ends a sentence. A Find command that overcomes these limitations is given in Section 2.10.3.

?%x Match Any Character Except x

This 3-character construct matches any character except the final character of the construct: any character that x would not normally match. Thus, a?%bc matches aac, acc, adc, and so forth, but not abc.

?)x Match 0 or More of the Character x

This construct matches the character that follows, any number of times that it appears. Thus, a?)bc matches ac, abc, abbc, and so on. SOS matches the first such string it encounters.

This construct cannot be used in isolation: the command F?)x(ESC) is illegal, and results in the error message ILLEGAL SEARCH STRING GIVEN. Such a construct, if legal, would match every line.

?1x Match 1 or More of the Character x

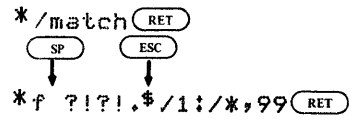
This construct is similar to the one described above, except that it matches one or more occurrences of the given character. Thus, a?1bc matches abc, abbc, abbbc, and so on, but not ac. SOS matches the first such string it encounters.

?9 Match Any Alphanumeric Character

This construct matches any character in the ranges A-Z, a-z, and 0-9.

?! Match Any Letter (A-Z, a-z)

This construct matches any uppercase or lowercase letter. You could use it, for example, to find all lines in a file that contain the strings Mr., Ms., or Dr.:



This Find command prints all lines that contain a space followed by two letters (of either case), followed by a period.

?& Match Any Uppercase Letter (A-Z)

This construct matches any uppercase letter. For example, 101-?& matches 101-A, 101-B, and so on, but not 101-7 or 101-f.

?2 Match Any Lowercase Letter (a-z)

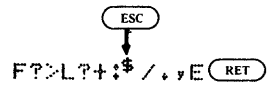
This construct matches any letter in the range a-z. For example, (?2) matches (i), (j), and so on, but not (A) or (4).

?+ Match Any Decimal Digit (0-9)

This construct finds a number. For example, ?+?+?+-A matches 101-A, 203-A, and so on, but not DEF-A.

?> Match Beginning or End of Line

This construct is useful in searching for a string that begins or ends a line. For instance, suppose you want to search a MACRO program for all labels of the form Ln:. You do not want to see lines in which these labels are referred to. Since labels occur at the beginning of lines, the following Find command performs the desired search.



Here the ?> construct matches the beginning of a line, and the ?+ construct matches any digit.

?7c Match Internal Representation of c

This construct matches the character that SOS uses internally to represent the construct c. These internal representations are given in Table 2-8. For example, suppose you need to search for the character CTRL/F. As Table 2-8 shows, SOS uses CTRL/F internally to represent the construct ?& (match any uppercase letter). Thus, to locate CTRL/F, use the combined construct ?7?&.

The use of the ?7 construct should be rare. Normally, in the infrequent instance when you need to search for an SOS internal representation character, it is easier to disable special pattern-matching and to search for the character directly. That is, type /MATCH:RET (note the colon) and then issue a Find command such as fCTRL/F ESC RET.

The only time you are required to use ?7 is when you must use other special pattern-matching constructs in a search. In that case, you cannot first disable special pattern-matching. One such case would be a Substitute command that searches for (CTRL/F) and then uses it in a Substitute-string construct in the newstring argument (see Section 2.10.2).

2.10.2 Substitute-String Constructs

When you use find-string constructs in the oldstring of a Substitute command, SOS keeps track of the first, second, ..., nth strings matched within each line by these constructs. The substitute-string constructs let you manipulate the strings matched by the special find-string constructs, without knowing the exact contents of the matched strings.

Use these two constructs in substitute strings, that is, in the newstring arguments of Substitute commands.

NOTE

Do not attempt to use substitute-string constructs to refer to a string matched by ?> (beginning or end of line). A beginning or end of line thus matched is not included in SOS's count of matched strings.

?*n?* Substitute nth String Matched

This special construct gives you access to the first, second, ..., nth strings matched within a line by special find-string constructs. One possible use of this feature is in exchanging arguments of functions. For example, if you want to change all occurrences of

```
FUNCTION(a,b)
```

to:

```
FUNCTION(b,a)
```

for any arguments a and b, you can issue the following Substitute command:

```
*/MATCH(RET)
                                  (ESC)                    (ESC)
*SFUNCTION(?)?/,?)?/)$FUNCTION(?*2?*,?*1?*)$/1:/*(RET)
```

?" Substitute Next String Matched

This construct is similar to the previous one. When used in a substitute string, it specifies substituting the next string in the series of strings matched by find-string constructs. Thus, if you use the constructs ?*3?*?" in a Substitute newstring, SOS substitutes the third and fourth strings matched within the line by find-string constructs.

2.10.3 Combining Find-String Constructs

Find-string constructs can be combined in a Find or Substitute command to give even greater versatility to SOS's pattern-searching features.

Perhaps the most useful Find-string construct to combine with others is ?%x (match any character except x). For example:

?%?9 matches any character that is not alphanumeric.

?%?+ matches any character other than a number.

?%?2 matches any character except a lowercase letter.

More than two Find-string constructs can be combined. For example:

?%?1?+ matches anything but one or more digits. This combined construct could be useful in searching a data file in which much of the text is multidigit numbers.

?1?%?< matches one or more of any character except spaces or tabs.

?%?7?& matches any character except the internal representation of the construct ?& (that is, anything but (CTRL/F)).

SOS does not consider the beginning or end of a line a character. Thus, the beginning or end of a line is matched by all of the following combined constructs (among others):

?%?/ anything that is not a character

?%?: not a separator

?%?> not a space or tab

?%?9 not alphanumeric

?%?! not a letter

?%?+ not a number

For example, you can search for all occurrences of the word "can" with the following combined construct. This Find command works if "can" begins or ends a sentence (assuming the EXACT switch is not on), or if it begins or ends a line.

(ESC)
↓
f?%?!can?%?!$/1:/,99(RET)

2.11 SPECIFYING RANGES BY CONTENT

For any Edit-mode command that requires a range of line numbers as an argument, you can implicitly specify the range by text content.

For example, instead of requesting that SOS print the lines numbered from 150 through 220, you can request that SOS print all lines from the line in which the string ABC occurs through the line in which the string DEF occurs. (You can do this for any SOS command, not just the Print command.)

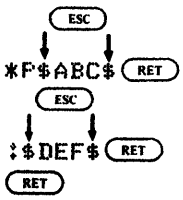
A common way to specify a range of lines is to use the following form:

position:position

To specify a range by content, replace either or both of the position specifications with arguments in the following form:

ESC string ESC RET

Terminate the command with a final RET. For example, to issue the Print command cited above, type:



Note that three carriage returns are required: one each to terminate the two content specifications, and a final one to terminate the Print command. Note also that a colon separates the two halves of the content-specified range.

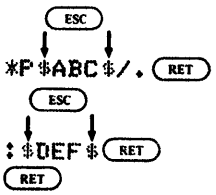
Using this kind of content specification is equivalent to embedding one or two Find commands within whatever command you are issuing. In executing the Print command given above, for instance, SOS performs a search for the string ABC as soon as you type the first RET. The search starts at the current line and proceeds to the end of the file. If SOS does not find the string, it prints an error message immediately after you type the first RET.

Assuming ABC is found, SOS performs a second search, starting at the position of ABC, when you type the second RET (terminating the range). Again, SOS prints an error message if it fails to find DEF. When you type the final RET, SOS performs the Print command, using the range of lines it has located in its two searches.

Normally, SOS starts its search for a content-specified line at the current line. However, you can specify a range over which SOS is to search with the form:

ESC string ESC range RET

For example, suppose the current line is beyond the first line on the current page that contains the string ABC. Without resetting the current line, you can effect the Print command given above by specifying /. as the range for SOS's content-specified search:



Content specification is more general still. After the range you can give options that specify:

- An exact pattern match (upper- and lowercase)
- A quasi-Decide mode
- The nth occurrence of the given string.

In other words, content specification has options that give it all the generality of a Find or Substitute command.

The full syntax of a content specification is as follows:

(ESC) [string] (ESC) [range] [{,D / ,N / ,n}] [,E] [,-] (RET)

where:

string
Specifies the pattern SOS is to search for. Note that string is optional. If you omit it, SOS uses the string last used in the same position in a content specification.

range
Specifies the range over which SOS is to search for the string. If you omit range, SOS searches from the current line to the end of the file.

,D
Tells SOS to enter a quasi-Decide mode. SOS prints the first line containing the string, then prompts with ?. You type Y, y, or (SP) if the line printed is the line you intend for the content specification. SOS then continues with the content-specified command. Type (DEL) to reject the line. SOS then prints the next line in the range that contains the string, and again prompts with ?. If you reject all the lines in the range that contain the string, SOS prints an error message and terminates the content-specified command.

,N
Tells SOS to enter the same quasi-Decide mode, but instead of printing each line found, SOS prints only the line number, and prompts with ?. Type Y, y, or (SP) to accept the line; type (DEL) to reject it.

,n
An integer: tells SOS to use the nth occurrence of the string within the range. Note that the meaning of this option differs from its meaning in the Find command. In Find, ,n tells SOS to print the next n occurrences of a string, not its nth occurrence.

,E
Tells SOS to accept only an exact pattern match in its search for the string. Upper- and lowercase letters must match. Note: the EXACT switch, which controls exact pattern matching in the Find and Substitute commands, is also effective in content specifications. Turning on the EXACT switch has the same effect as specifying the ,E option.

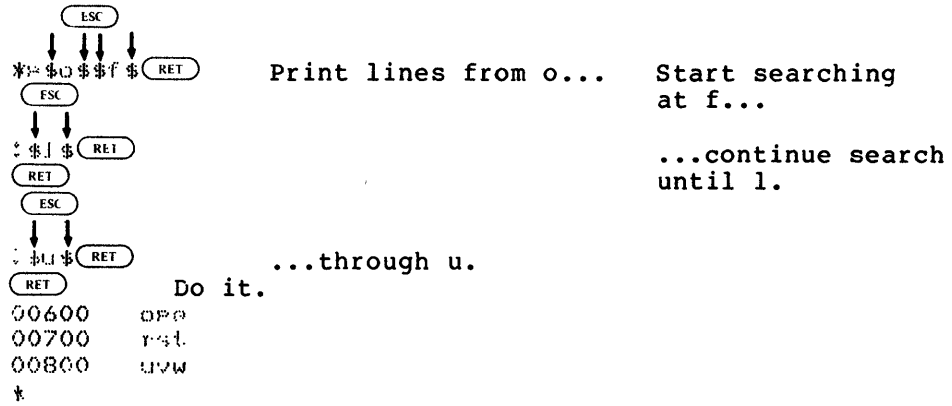
,-
Tells SOS to use the first line in which the specified pattern does **not** occur.

Content specification has still another sort of generality. The range that you can give in a content specification can itself be content-specified; the range in that content specification can also be content-specified. The maximum nesting depth is three levels.

For example, a file contains the following lines:

```
00100   abc
00200   cde
00300   fgh        ]
00400   ijk        |
00500   lmn        |
00600   opq   ]    |
00700   rst   |    |
00800   uvw   ]    |
00900   xyz        |
01000   123        ]
01100   234
01200   uuo
01300   890
```

As indicated by the brackets, the user wants to print the lines between the first occurrence of o and the next occurrence of u, with the searches constrained to the range between the first occurrence of f and the next occurrence of l. The following two-level, content-specified Print command performs this task:



2.12 SOS SPECIAL CHARACTERS

Table 2-9 shows all the nonalphabetic characters that have special meaning to SOS. The meaning of some characters depends on the context (for instance, ESC can have one of several meanings, depending on the mode SOS is operating in).

Table 2-9
SOS Special Characters

Character	Meanings
.	Edit mode: Move Position command. In position specifier: means current line or current page.
=	Edit mode: Give Parameter command. In Copy command: means text is copied from another file.
+	In position specification: signifies a positive line-number offset. Special pattern-matching construct: match any decimal digit.
@	Edit mode: Command File command.
/	Edit mode: Set Parameter command. In Copy command: /C means enter Copy-file mode. At DCL level: marks beginning of a qualifier. In position specifier: separates line number and page number. Special pattern-matching construct: match any character.
^	In position specifier: means first line on a page, or first page in a file.
*	In position specifier: means last line on a page, or last page in a file. Special pattern-matching construct: substitute the nth string matched by a special construct.
:	In End, Save World, List commands: signals beginning of a filespec. In range specification: separates starting and ending positions. In Set Parameter command: signals beginning of parameter value. Special pattern-matching construct: match any separator.
!	In range specification: signals relative number of lines. Special pattern-matching construct: match any letter (A-Z, a-z).
,	In Input, Replace commands: signals new line-number increment. In Copy, reNumber, Transfer commands: separates arguments. In End, Find, List, Print, Substitute, Save World, eXtend commands: separates command options (,A ,D ,E ,F ,N ,P ,S ,-).

(continued on next page)

Table 2-9 (Cont.)
SOS Special Characters

Char-acter	Meanings
-	In Find command and in content specification (all commands): used as a command option. Tells SOS to find the first line **not** containing the specified pattern. In position specification: signifies a negative line-number offset.
;	In Input, Replace commands: signals temporary line-number increment for this command only.
?	Default flag character for special pattern-matching.
$	Default logical escape character.
<	Special pattern-matching construct: match a space or tab.
>	Special pattern-matching construct: match begining or end of line.
%	Special pattern-matching construct: match any character except the given one.
)	Special pattern-matching construct: match 0 or more of the given character.
&	Special pattern-matching construct: match any uppercase letter.
"	Special pattern-matching construct: substitute the next sequential string matched by a special construct.
ESC	Edit mode: print previous line. Input mode: return to Edit mode. Alter/insert mode: return to Alter mode. Decide Alter/insert mode: return to Decide Alter mode. Decide mode: illegal. In Find, Substitute commands: delimits text strings for search and substitution.
BSP	Alter mode: move intraline pointer back one character.
RET	Edit mode: print next line. Alter mode: print rest of line, finish intraline edit for this line. Alter/insert mode: print rest of line, finish intraline edit for this line. Decide Alter mode and Decide Alter/insert mode: print rest of line, return to Decide mode.

(continued on next page)

Table 2-9 (Cont.)
SOS Special Characters

Character	Meanings
DEL	Alter mode and Decide Alter mode: move intraline pointer back one character. Decide mode: no, do not make the change. All other modes: delete previous input character.
SP	Alter mode and Decide Alter mode: move intraline pointer forward one character. Decide mode: yes, make the change.
TAB	Alter mode and Decide Alter mode: move intraline pointer to the end of the line.
CTRL/U	Alter mode and Decide Alter mode: discard edits performed so far and place intraline pointer at beginning of line. Alter/insert mode and Decide Alter/insert mode: take CTRL/U as text and insert it into the text line.

2.13 SOS INTERNALS

This section discusses SOS temporary files and the format of SOS text lines and page marks. You can make the most efficient use of the editor if you know what SOS does with its temporary files, and how these files are related to the input and output files you specify.

2.13.1 SOS Temporary Files

Whenever you are using the editor, you have either one or two SOS temporary files in your directory. Their names are SOSxnn.TM1 and SOSxnn.TM2 (x is the controller number and nn is your terminal number).

Two events within an editing session affect the status of these temporary files: your issuing a Save World command, and SOS performing a "wraparound." To understand the wraparound operation, consider the process of editing a file.

As you edit a file, SOS reads lines into an internal edit buffer[1] and writes them out to the temporary file SOSxnn.TM1. Changes you make while editing are incorporated into this file. This process of passing text through the edit buffer to the temporary file continues as long as your editing does not refer to a line before the start of the edit buffer. (The position of the start of the edit buffer is marked by the parameter LOCATION, described in Section 2.9.1.)

1. The edit buffer is an in-image storage area whose size is determined when SOS is linked; a typical size is 1500 bytes.

When you do call for editing on a line before the start of the edit buffer, SOS performs a wraparound. This means that SOS copies the rest of the file (from the current position to the end) to SOSxnn.TM1. At this point, SOSxnn.TM1 is a complete copy of your original file, with all edits to date incorporated. SOS then "rewinds" SOSxnn.TM1: it returns to the beginning of the temporary file and starts reading from it.

Thereafter, SOSxnn.TM1 serves as the input file for further editing. SOS closes your original file, and does not read it again.

After rewinding, SOS creates a second temporary file, SOSxnn.TM2, to which it writes lines passed through the edit buffer. SOS continues this pass-through operation until it finds the line referred to in your command.

The next time wraparound is necessary, the temporary files switch roles. SOS rewinds both files, and uses SOSxnn.TM2 (which now contains the most up-to-date copy of the file) for input. SOS passes text lines through the edit buffer and writes them to SOSxnn.TM1. Neither file is closed.

When you issue a Save World command, the temporary files again effectively switch roles. The temporary file being used for output is renamed to the file specifier given -- or provided by default -- in the Save World command. SOS closes this renamed file, and then opens it (read-only) for input. The temporary file that was used for input is rewound and thereafter is used for output.

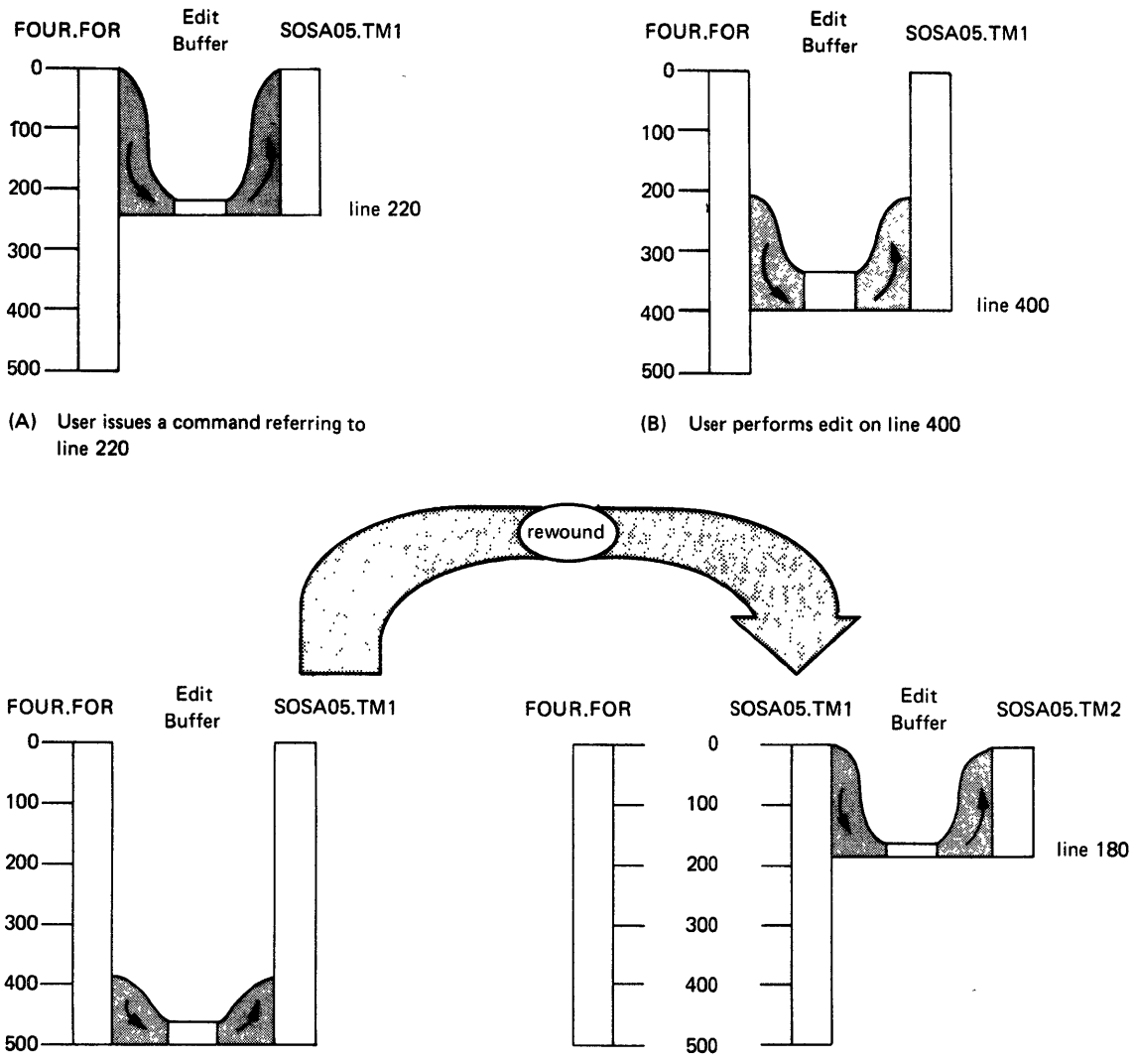
If you give another Save World command, or if you give an End command without a new file specification, SOS deletes the file created by the last Save World command before renaming the output temporary file.

If you try to recover information from temporary files as described in Section 2.7.4, you may encounter some errors when SOS tries to read data near the end of the file. The errors occur because the file was never closed properly and has no end-of-file mark. You can simply delete the erroneous line(s).

2.13.1.1 **Temporary File Example** - Figure 2-6 depicts an example of the wraparound process. A user named Brand is working at terminal number 5. Brand edits a 500-line file (a FORTRAN source file for a Fourier-analysis program) named FOUR.FOR. In illustration (a) in Figure 2-6, Brand gives a Find command that moves the current line to line 220. At this point, the file SOSA05.TM1 is a copy of FOUR.FOR from the beginning to line 220.

If Brand typed =LOCATION (RET) at this point, he would learn that the start of the edit buffer is at line 200.

Illustration (b) in the figure shows what happens when Brand performs some editing on line 400. More of FOUR.FOR is passed through the edit buffer and written to SOSA05.TM1.



(A) User issues a command referring to line 220

(B) User performs edit on line 400

(C) User issues a command referring to line 180, which is before the start of the edit buffer. Therefore, SOS performs a wraparound.

Figure 2-6 Temporary Files in a Sample Editing Session

The start of the edit buffer is now at line 340. The lines are shorter in this part of the source file, so more lines fit into the fixed-length edit buffer.

Illustration (c) shows the wraparound process. Brand issues a Find command that matches a string in line 180. SOS copies the rest of the file to SOSA05.TM1, rewinds the file, and uses it as the input file. SOS then creates SOSA05.TM2; this file receives text passed through the edit buffer up to line 180.

2.13.1.2 **Using SOS Efficiently** - In Read-only mode, SOS does not write an output file. The pass-through and wraparound file-writing described above does not occur. For this reason, SOS uses far less system resources in Read-only mode. This is why it is best to use Read-only mode routinely whenever you only need to examine a file. (You might use symbol assignment to define a command such as READ:="EDIT/SOS/READONLY".)

The most efficient way to use SOS in Edit mode is to avoid wraparound. You can do this by editing a file sequentially from beginning to end. If you never refer to a line earlier in the file than the position marked by LOCATION, SOS does not wrap the file. You minimize the need for costly and time-consuming file copying.

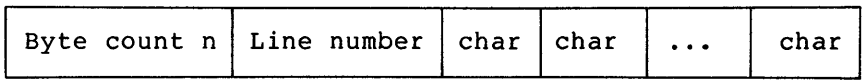
SOS's wraparound behavior explains why writing an output file without line numbers (ES or ET commands) requires more processing time than leaving the line numbers in place (as mentioned in Section 2.8.4). Suppose, for example, that the current line is in the middle of the file when you issue an ES command. SOS passes text through the edit buffer until end of file; rewinds both temporary files; then passes the entire file through a second time, omitting line numbers. SOS then renames the output temporary file to the output name, destroys the other temporary file, and exits.

If instead of an ES command you issue an E command, SOS need not pass the file through the edit buffer a second time. After first passing through to end of file, SOS renames the output temporary file to the output name, destroys the other temporary file, and exits.

2.13.2 Format of Text Lines and Page Marks

When stored in a line-numbered file, a line of text generated by SOS is a VAX-11 RMS record. Its type is variable with fixed-length control. The SOS line number comprises the fixed-control access field of the VAX-11 RMS record.

The text line has the following format:



The byte count, a 16-bit integer, is 2 more than the number of characters in the text line. The line number is also a 16-bit integer. The text consists of 500 or fewer characters (represented above by char) that do not include the characters DEL or NU. No RET or LF is present to terminate the text line; its end point is implied by the byte count.

An SOS page mark is represented in a file by the following special case of the general format given above:

When SOS displays such text lines, it prints the line number as a 5-digit ASCII number; prints a TAB; prints the n-2 characters of text; and adds a RET and LF to the line.

SOS does not print a page mark as such. Instead, when it displays text on a page other than the current page, it prints PAGE n. first.

CHAPTER 3

BATCH-ORIENTED TEXT EDITOR (SLP)

3.1 INTRODUCTION TO SLP

SLP is a batch-oriented editing program used for source file maintenance.[1] SLP allows you to update (delete, replace, add) lines in an existing file. Furthermore, SLP gives you a record of editing changes. The SLP command file provides a reliable way to duplicate the changes made to a file, at a later time or on another computer system.

Input to SLP consists of (1) a correction input file that you want updated, and (2) a command file containing text lines and edit command lines that specify the update operations to be performed. SLP locates lines to be changed by means of "locators" (sequence numbers or character strings).

SLP output is a listing file and an updated copy of the correction input file. SLP provides an "audit trail" that helps you keep track of the update status of each line in the file. The audit trail is shown in the listing and is included permanently in the output file. When a given file is updated with successive versions of an SLP command file, you can use different audit trails to differentiate among the changes made at different times.

SLP output qualifiers let you truncate lines, create or suppress an audit trail, eliminate an existing audit trail, and specify the length and beginning position of the audit trail.

3.2 RUNNING SLP

Run SLP by submitting a command file for processing. The command file contains the EDIT/SLP initialization line, SLP edit commands, and a DCL $EXIT command. You submit the command file by using either the SUBMIT command or the @ (Execute Procedure) command.

You must create the command file before running SLP. The interactive text editor SOS is usually used to create SLP command files.

In the following example, SLP is run by submitting the command file UPDATE.COM. (The required contents of such a command file are detailed in Section 3.3.2.) Note that, since the file type of the command file is COM, the type can be omitted on the DCL command line.

```
$ @UPDATE (RET)
```

When SLP finishes its processing, the DCL prompt is issued.

---

1. SLP originally meant "source language input program." Today, interactive text editors like SOS are used more often for file input.

3.3 SLP INPUT AND OUTPUT FILES

SLP requires two types of input files: a correction input file and a command file. The correction input file is the source file you want to update. The command file consists of an initialization line, SLP edit commands that indicate how the file is to be changed, and the DCL command $EXIT.

The listing file is a copy of the output file with sequence numbers added; it shows the changes SLP makes to the correction input file. The output file is the permanently updated copy of the input file. SLP output consists of a listing file and an output file. Figure 3-1 shows the relationships among the SLP input and output files. Later sections show the contents of the various files in this example.

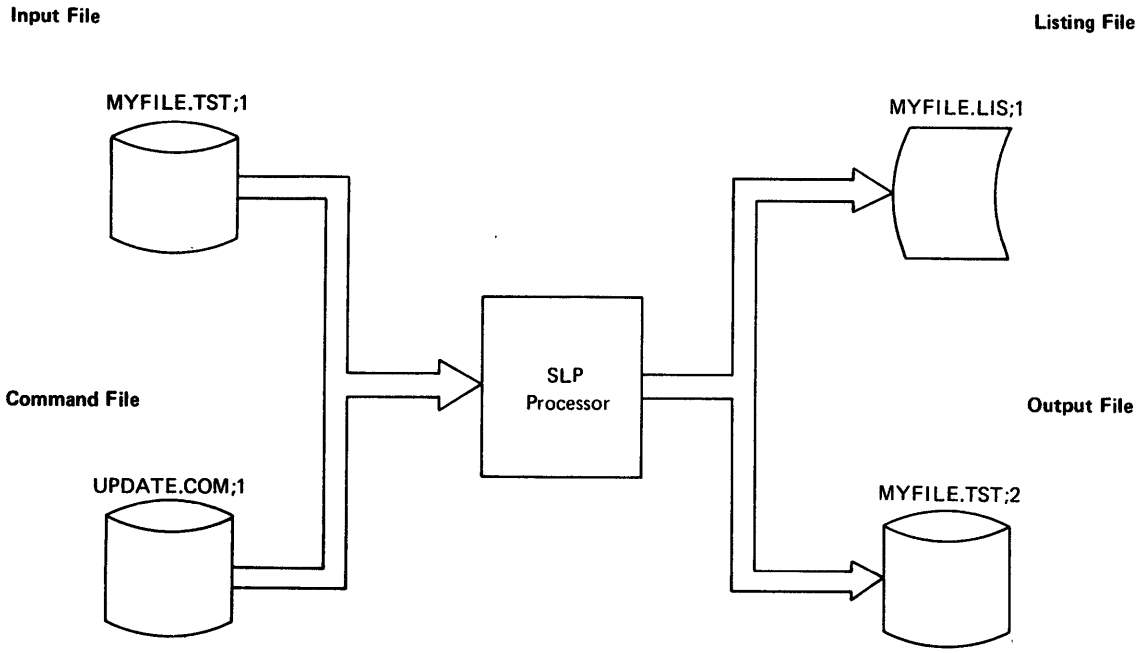


Figure 3-1 Input Files and Output Files Used During SLP Processing

3.3.1 The Correction Input File

The correction input file is the file to be updated by SLP. It can contain any number of lines of text. When SLP processes the correction input file, it makes the changes specified by SLP edit commands and marks these changes with an audit trail (discussed in Section 3.7) in the output file.

3.3.2 The SLP Command File

The SLP command file is a VAX/VMS procedure file that contains SLP edit commands. It consists of:

1. An **initialization line** that tells SLP what files to process. This line has the general form:

   EDIT/SLP[/qualifiers] input-file

   The optional qualifiers are described in Section 3.6. This initialization line must be the first line in the command file.

2. SLP **edit command lines** that define changes to the input file (see Section 3.5.1).

3. Input lines. These lines of text are inserted into the output file, either as new lines or to replace old lines in the file.

4. An SLP terminator. A single slash in column 1 causes SLP to complete its processing of the files.

5. A command terminator -- the DCL $EXIT command.

The example below shows a command file. The numbered lines correspond to the items listed above.

```
$EDIT/SLP MYFILE.TES                                    (1)
-3                                                      (2)
INSERT THIS LINE AFTER LINE 3                           (3)
-4,4                                                    (2)
DELETE LINE 4 AND REPLACE IT WITH THIS LINE             (3)
/                                                       (4)
$EXIT                                                   (5)
```

3.3.3 The SLP Listing File

The SLP listing file shows the updates made to the source file. Each line in the listing file is numbered in sequence. Updates are marked by means of an audit trail (unless you specify the qualifier that suppresses audit-trail generation). Section 3.4 contains an example of a listing file.

A listing file is generated by default. You can suppress the listing file by using the /NOLISTING qualifier; see Section 3.6.1.

3.3.4 The SLP Output File

The SLP output file is the updated input file. All of the updates specified by SLP edit commands are inserted in this file. A default audit trail, unless suppressed, is applied to new or changed lines. The numbers generated by SLP for the listing file do not appear in the output file.

An output file is generated by default. You can suppress the output file by using the /NOOUTPUT qualifier; see Section 3.6.2.

3.4 HOW SLP PROCESSES FILES

This section uses an example to show how SLP processes files. The following test file is named MYFILE.TST.

```
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE
TEN
```

This file is to be updated under the control of the following command file, named UPDATE.COM:

```
$EDIT/SLP/AUDIT_TRAIL:(POSITION=50) MYFILE.TST
-3
INSERT THIS LINE AFTER LINE 3
-4,4
DELETE LINE 4 AND REPLACE IT WITH THIS LINE
/
$EXIT
```

Figure 3-1 represents the process of updating this file by issuing the command @UPDATE. Below is the listing file (MYFILE.LIS) that results from this operation.

```
 1.  ONE
 2.  TWO
 3.  THREE
 4.  INSERT THIS LINE AFTER LINE THREE                ;**NEW**
 5.  DELETE LINE 4 AND REPLACE IT WITH THIS LINE      ;**NEW**
 6.  FIVE                                             ;**-1
 7.  SIX
 8.  SEVEN
 9.  EIGHT
10.  NINE
11.  TEN
```

The audit trail shows the new lines (;**NEW**) and indicates where lines have been removed (;**-1). In this case, a new line has been added after line 3, and line 4 has been replaced, causing all subsequent lines to be renumbered. The /AUDIT_TRAIL qualifier in the initialization line indicates that the audit trail is to begin at the next tab stop after column 50.

To process the files, SLP writes each line from the correction input file into the output file until it reaches a line to be modified, as requested in the command. When SLP reaches a line to be modified, it makes the indicated modification, notes the change in the audit trail, and then continues writing lines to the output file, in sequence, until it encounters another command or reaches end of file.

3.5 USING SLP

This section contains information on how to specify SLP edit commands and update files using SLP.

3.5.1 Specifying SLP Edit Commands

SLP edit commands let you update source files by adding, deleting, and replacing lines in a file. SLP edit commands are marked by certain characters that SLP interprets as operators. This section first describes the operators SLP interprets specially, then describes the general form for specifying SLP edit commands.

3.5.1.1 SLP Operators - SLP interprets the following characters specially when you enter one of them as the first character of an input line: the minus sign (-), the backslash (\), the percent sign (%), the at sign (@), the slash (/), and the less-than character (<). Table 3-1 lists each of these operators and the functions they perform.

The at sign (@) operator tells SLP to read further input from a another command file. This second command file can contain only SLP edit commands and new text lines.

The less-than character (<) is the escape character that lets you enter characters in the command file (in column 1) that SLP otherwise would interpret as operators. For example, </ hides the slash character from SLP, thereby enabling you to enter the slash into the output file without terminating the SLP edit session. You can use the less-than character as an escape character for all SLP operators listed in Table 3-1 (including itself).

Table 3-1
SLP Operators

Operator	Function
-	First character of an SLP edit command
\	Suppress audit-trail generation
%	Reenable audit-trail generation
@	Invoke a further command file for SLP processing
/	Terminate the edit session
<	Escape character

3.5.1.2 General Form of the Edit Command - The general form of the edit command is as follows:

```
-locator1[,locator2][,/audittrail/][;comment]
inputline
      .
      .
      .
```

Arguments:

- (minus sign)
  Specifies that this is an SLP edit command line.

locator1
Is a line locator that causes SLP to move the current line pointer to a specified line. If only locator1 is specified, the current line pointer is moved to that line and SLP reads the next line in the edit command file. This field can be specified using any of the locator forms described below.

locator2
Is a line locator that defines a range of lines (that is, the range beginning with locator1 and ending with locator2) to be deleted or replaced. This field can be specified using any of the locator forms described below.

/audittrail/
Is a character string used to keep track of the update status of each line in the file. This audit trail is used to mark new or replaced lines in the file until the audit trail is either changed or suppressed. This argument must be delimited by slashes (/).

inputline
Is a line of new text to be inserted into the file immediately following the current line. You can enter any number of input lines.

;comment
Is an optional comment. SLP ignores any text after a semicolon.

All fields in the command line are position-dependent; commas must be specified.

The locator fields can take one of the following forms:

$$\left\{\begin{array}{c}/\text{string}[...\text{string}]/\\ \text{number}\end{array}\right\}\ [+n]$$

Arguments:

string
Is a string of ASCII characters. SLP locates the next line in which string exists and moves the current line pointer to that line. If the locator is specified in the form /string...string/ (that is, two different strings of characters separated by three periods), SLP locates the line in which the first character string is followed by the second character string, regardless of what characters may be in between them.

number
Specifies a sequence number to which the current line pointer is to be moved. The largest sequence number that can be specified is 9999.

n
Specifies a decimal value used as an offset from the line specified by the locator.

. (period)
Indicates the current line.

All forms of the line locator can be specified interchangeably in a command line.

SLP can only edit files sequentially. Once the current line pointer moves past a given line in the file, it cannot be returned to that line.

3.5.2 Updating Source Files Using SLP

This section describes how to use the SLP edit command to add, delete, and replace lines in a file. The section also includes a procedure for generating a numbered listing for use in editing source files by sequence number.

3.5.2.1 **Generating a Numbered Listing** - SLP processes input by sequence number. However, sequence numbers appear only in the listing file; they are not written to the output file.

To use SLP effectively, obtain an up-to-date numbered listing for use when you prepare the command file. Numbered listings generated by other programs (such as SOS and the MACRO assembler) will not necessarily be useful in preparing an SLP command file. Generate an SLP numbered listing by submitting a command file in the following form:

```
$EDIT/SLP/NOOUTPUT/LISTING:list-file] input-file
/
$EXIT
```

Here list-file is the name you optionally assign to the listing file that SLP produces, and input-file is the specification of the file whose lines are to be numbered. The slash (/) tells SLP to begin processing the files. SLP generates a numbered listing file, but does not produce an output file.

3.5.2.2 **Adding Lines to a File** - The SLP edit command for adding lines to a file contains only one locator field. Its form is given below.

```
-locator[,,/audittrail/][;comment]
```

where (as in Section 3.5.1.2), locator has the following form:

```
{/string[...string]/}
{      number       }  [+n]
{        .          }
```

If number is specified, SLP inserts new line(s) after the line specified by sequence number. Any lines you enter are inserted as lines in the file.

If string is specified, SLP locates the next occurrence of string in the file and moves the current line pointer to the line containing string. Any input lines following the command line are then added to the file.

If you specify +n, SLP moves the current line pointer n lines beyond the line specified in the locator field and then adds any new input lines to the file.

The example below shows how to add lines to a file. The correction input file consists of the following lines:

```
ABC
DEF
GHI
KLM
123456789
456
789
CBA
XYZ
987
```

The input file consists of the following commands and text lines:

```
$EDIT/SLP MYFILE.TST
-/123/
INSERT THIS LINE AFTER LINE 5
/
$EXIT
```

SLP processing generates the following listing file:

```
 1.  ABC
 2.  DEF
 3.  GHI
 4.  KLM
 5.  123456789
 6.  INSERT THIS LINE AFTER LINE 5       ;**NEW**
 7.  456
 8.  789
 9.  CBA
10.  XYZ
11.  987
```

SLP has applied sequence numbers to the lines and added an audit trail to the line following line 5, where SLP found the first occurrence of the string 123.

The next example uses the same correction input file and the following new command file:

```
$EDIT/SLP MYFILE.TST
-/DEF/+2
THIS IS NEW TEXT
/
$EXIT
```

SLP processing generates the following listing file:

```
 1.  ABC
 2.  DEF
 3.  GHI
 4.  KLM
 5.  THIS IS NEW TEXT                    ;**NEW**
 6.  123456789
 7.  456
 8.  789
 9.  CBA
10.  XYZ
11.  987
```

Again, SLP has numbered the lines in sequence; this time the new input line is inserted two lines beyond the line containing the first occurrence of the string DEF.

3.5.2.3 **Deleting Lines from a File** - The SLP edit command for deleting lines from a file contains two locator fields. Its form is given below.

```
-locator1,locator2[,/audittrail/][;comment]
```

where locator1 and locator2 can be any of the forms of the locator fields described above. The first field, locator1, specifies the line where SLP is to begin deleting lines; locator2 specifies the last line to be deleted. SLP deletes all lines from locator1 through locator2, inclusive.

The example below shows how to delete lines from a file using SLP. The correction input file consists of the following lines:

```
ABC
DEF
GHI
KLM
123456789
456
789
CBA
XYZ
987
```

The command file for this example is as follows:

```
$EDIT/SLP  MYFILE.TST
-/1...9/,/XYZ/
/
$EXIT
```

SLP processing generates the following listing file:

```
1.  ABC
2.  DEF
3.  GHI
4.  KLM
5.  987                                    ;**-5
```

In this example, the ellipsis (...) is used to abbreviate the larger string 123456789. SLP searches for the first occurrence of the string 1 and the first occurrence of the string 9 on the line, assuming these two strings bracket a larger string, in this case, the string 123456789. SLP begins deleting lines at this line and continues deleting lines until it deletes the last line, specified by the string XYZ. SLP applies the audit trail count of the lines it deleted to the next line in the output file.

Using the same correction input file, this example shows how to delete a single line using the period locator. The command file for this example is as follows:

```
$EDIT/SLP MYFILE.TST
-/DEF/,.
/
$EXIT
```

SLP processing generates the following listing:

```
1.  ABC
2.  GHI                                    ;**-1
3.  KLM
4.  123456789
5.  456
6.  789
7.  CBA
8.  XYZ
9.  987
```

SLP moves the current line pointer to the line containing the string DEF and then finds the period as the second locator field. Since the second locator field is specified, SLP interprets the edit command as a delete operation and deletes the line containing DEF.

3.5.2.4 **Replacing Lines in a File -** A replacement is a deletion followed by new text. The number of lines deleted need not match the number of lines added. To replace lines in a file, use the full 2-locator command form, as in the delete command. The first line locator field specifies the first line to be deleted. The second line locator field defines the last line in the range to be deleted, and, for replacement operations, the line where new text is to be inserted.

For example, the command -4,.+4 instructs SLP to move the line pointer to line 4 and replace line 4 and the next four lines (as represented by .+4) with new input lines that immediately follow the command line. This command is equivalent to -4,8.

The example below shows how to delete lines from a file and replace them with new lines. The correction input file consists of the following lines:

```
ABC
DEF
GHI
123456789
BCN
CRB
BUR
```

The command file is as follows:

```
$EDIT/SLP MYFILE.TST
-2,.+1
NEW LINE 2
NEW LINE 3
/
$EXIT
```

SLP processing generates the following listing file:

```
1.  ABC
2.  NEW LINE 2                             ;**NEW**
3.  NEW LINE 3                             ;**NEW**
4.  123456789                              ;**-2
5.  BCN
6.  CRB
7.  BUR
```

3.6 SLP QUALIFIERS

SLP qualifiers control the generation and format of the listing file and the output file. You can use them to control the audit trail and output options associated with these files. The following sections describe the SLP qualifiers and the functions each performs.

3.6.1 /[NO]LISTING[:file-spec]

The listing file that SLP generates by default has the same file name as the correction input file; its file type is LIS. You can request a different specification for the listing file by using the /LISTING:file-spec qualifier.

To suppress the listing file, specify /NOLISTING.

3.6.2 /[NO]OUTPUT[:file-spec]

By default, SLP generates an output file with the same file name and file type as the correction input file. Its version number is higher by 1 than the highest version number existing for the input file name and type. You can give a different specification by using the /OUTPUT:file-spec qualifier.

To suppress the output file, specify /NOOUTPUT.

3.6.3 /[NO]AUDIT_TRAIL[:(POSITION:pos,SIZE:len)]

This qualifier lets you suppress audit-trail generation, or specify the beginning position and length of the audit trail. The default is to generate an audit trail 8 characters long, starting in column 80 -- that is, /AUDIT_TRAIL:(POSITION:80,SIZE:8).

The maximum allowed value for the length parameter is 16. If you give either the position or length parameters, you must give both.

The audit trail starts at the first tab stop after the position given (or defaulted) for the /AUDIT_TRAIL qualifier. Tab stops are set every 8 columns. The /AUDIT_TRAIL qualifier is discussed in more detail in Section 3.7.

3.6.4 /[NO]TAB_FILL

The qualifier /TAB_FILL causes SLP to insert tabs at the end of each text line containing an audit trail. The default is to fill such lines with spaces (that is, /NOTAB_FILL). Using /TAB_FILL saves disk space, because fewer tabs than spaces are required to fill the lines in both the output file and the listing file.

3.6.5 /[NO]TRUNCATE[:position]

This qualifier lets you truncate input lines to the given column position. The default is /NOTRUNCATE. If you specify /TRUNCATE but omit position, SLP uses the position given (or defaulted) for the /AUDIT_TRAIL qualifier.

3.7 THE AUDIT TRAIL

The audit trail is a device that allows you to keep track of changes made to a file with SLP. The qualifier /[NO]AUDIT_TRAIL lets you suppress the generation of the audit trail, or set its beginning position and length. With SLP edit commands, you can change the value (that is, the text) of the audit trail, and temporarily suppress and resume audit-trail processing.

To suppress audit-trail generation initially, specify the qualifier /NOAUDIT_TRAIL.

By default, the audit trail is initially turned on. Its text value is ;**NEW**. That is, the text ;**NEW** is placed starting in column 80 of each new or changed line.

3.7.1 Changing the Value of the Audit Trail

The following SLP edit command changes the value of the audit trail:

```
-,,/newtrail/
```

Here newtrail is the new value (text) of the audit trail. If the length of newtrail exceeds the length specified (or defaulted) for the /AUDIT_TRAIL qualifier, the audit trail is truncated to that length. (The default audit trail, ;**NEW**, is never truncated, even if you specify a length less that 8.)

The new audit trail is used in all subsequent lines changed.

3.7.2 Setting the Position and Length of the Audit Trail

To set the beginning position and the length of the audit trail, use the following qualifier:

```
/AUDIT_TRAIL:(POSITION:pos,SIZE:len)
```

The example below shows the use of this qualifier. The correction input file for this example follows.

```
ONE
TWO
THREE
FOUR
FIVE
```

The command file is as follows:

```
$EDIT/SLP/AUDIT_TRAIL:(POSITION:30,SIZE:10)  MYFILE.TST
-2,.+1,/;CHANGE001/
NEW LINE 2
NEW LINE 3
/
$EXIT
```

The following listing file results from SLP processing.

```
1.  ONE
2.  NEW LINE 2                    ;CHANGE001
3.  NEW LINE 3                    ;CHANGE001
4.  FOUR
5.  FIVE
```

3.7.3 Temporarily Suppressing the Audit Trail

You can temporarily suppress the audit trail in the output and listing files by using the backslash (\) operator. You can reenable audit-trail generation with the percent sign (%) operator, even if you initially specified /NOAUDIT_TRAIL.

Use the backslash in column 1 of the command file to suppress audit-trail generation. Use the percent operator in column 1 at some later point to resume the audit trail.

3.7.4 Deleting an Input Audit Trail

You can delete audit trails in the input file in one of the following ways:

- The default audit trail, ;**NEW**, is automatically deleted when the file is read with audit-trail processing enabled.
- Any other audit trail can be deleted by using the /TRUNCATE qualifier.

The qualifier /TRUNCATE[:position] truncates input lines at the given column position. If you give the /TRUNCATE qualifier but omit position, SLP uses the position given (or defaulted) for the /AUDIT_TRAIL qualifier. Set position at or before the start of the old audit trail you want deleted. Any trailing spaces or tabs after position are also deleted.

APPENDIX A

SUMMARY OF SOS FUNCTIONS AND FEATURES

Table A-1
SOS Modes of Operation

Mode	Prompt	Used for
Edit	*	Editing -- adding, changing text
Input	nnnnn	Typing in lines of text
Alter	nnnnn	Editing within a line
Alter/insert	(none)	Inserting text within Alter mode
Read-only	R*	Examining a file without changing it
Copy-file	C*	Searching, copying from another file
Decide	D*	Deciding case by case on substitutions
Decide Alter	nnnnn	(Submode) Like Alter mode
Decide Alter/insert	(none)	(Submode) Like Alter/insert mode

Table A-2
Edit-Mode Commands: Function and Format

Command	Function	Format[1]
Alter	Enter Alter mode for intraline, character-by-character editing.	A [range]
Copy	Copy a range of lines to another place within a file, or from another file. Optionally, enter Copy-file mode.	C position[=file-spec],range[,incr1[,incr2]] position=file-spec/C
Delete	Delete a range of lines.	D [range]
End	End SOS, return to DCL.	E [B] [Q] [S] [T] [:file-spec]
Find	Search for the occurrence of one or more specified strings of text.	F [[string] ESC [range] [,A] [,N] [,E] [,n] [,-]]
Help	Display the help package.	H [:n]
Input	Enter Input mode to insert lines of text.	I [position] [{ ,incr ; ;incr ;!n }]
Join	Join two text lines into one line.	J [position]
Kill Page Mark	Delete a page mark.	K /page
List	List a range of lines on the printer or to a file.	L [range] [,S] [{ ,P[:file-spec] ,F:file-spec }]
Mark	Insert a page mark.	M [position]
reNumber	Renumber a range of lines.	N [incr] [,[range] [,start]]
Print	Print a range of lines on the terminal.	P [range] [,S]
Replace	Delete a range of lines and enter Input mode.	R [range] [{ ,incr ;incr ;!n }]
Substitute	Replace one or more text strings with other string(s) in a range of lines. Optionally, enter Decide mode.	S [[oldstring ESC newstring] ESC [range] [,D] [,N] [,E]]
Transfer	Copy a range of lines to a new location and delete the original lines.	T position,range[,incr1[,incr2]]
Save World	Write a new file containing all the changes made so far.	W [B] [:file-spec]
eXtend	Enter Alter/insert mode to add text to the end of a line.	X [range] [,N]
Move Position	Reset the position of the current line.	. position
Give Parameter	Give the value of an SOS internal parameter or switch.	= parameter
Set Parameter	Reset an SOS parameter or switch.	/ parameter[:value]
Command File	Execute the SOS commands contained in a command file.	@ file-spec
RET	Print next line.	RET
ESC	Print previous line.	ESC

[1] The space after the single-character command is optional. Terminate each command with RET (except the commands RET and ESC).

Definitions for the Edit-Mode Commands

Uppercase letters represent literals that you type verbatim. (SOS accepts either upper- or lowercase letters.) Lowercase terms represent variables, defined below. The symbol => means "is defined as." Brackets ([ ]) indicate optional parts of the argument string (except in directory or filespec, in which one set of brackets must be typed). Stacked braces ({ }) mean to choose one of the enclosed expressions. Angle-brackets (< >) enclose descriptive terms. And (ESC) represents the escape key (or ALTmode, SELect or PREfix).

```
    file-spec =>[device:][[directory]]filename[.type[;version]]

    increment => <a positive integer line-number increment>

            l => <a line number> (less than 65536)

         line => { l } [± m]
                 { ^ }
                 { * }
                 { . }

            m => <an integer line-number offset>

            n => <a positive integer>

            p => <a page number>

         page => { p } [± q]
                 { ^ }
                 { * }
                 { . }

    parameter => <a legal SOS parameter or switch>

     position => { line [/page] }
                 {     /page    }

            q => <an integer page-number offset>

                 { position [{:position}]  }
                 {          [{    !n   }]  }
        range => {                        }
                 {       :position        }
                 {                        }
                 {          !n            }

        start => <a positive integer line number>

       string => string1[(RET)string2[(RET)string3[(RET)string4
                         [(RET)string5[(RET)string6]]]]]

        value => <a legal parameter value>
```

Table A-3
Meanings of Edit-Mode Command Options

Option	Meaning	Commands
,A	Enter Alter mode	Find
,D	Enter Decide mode	Substitute
,D	Enter quasi-Decide mode	Content specification (all commands)
,E	Require exact case match	Find, Substitute, content specification (all commands)
,F	List to disk file	List
,N	Type only number, not line	Find, Substitute, eXtend, content specification (all commands)
,P	Select printer	List
,S	Suppress line numbers	List, Print
,-	Find line in which specified pattern does not occur	Find, content specification (all commands)
,n	Find n occurrences of specified pattern	Find
,n	Use nth occurrence of specified pattern	Content specification (all commands)

Table A-4
Alter-Mode Commands

Command	Meaning
B	Do not print rest of line; recycle to beginning of line with edits incorporated.
[-][n]Cx...	Change next/previous n characters to given characters.[1]
[-][n]D	Delete next/previous n characters.
E	End Alter mode for this line; do not print line.
[-][n]Itext(ESC)	Insert text after/before pointer; use temporary increment n for new lines.
[-][n]Kx	Kill (delete) all until nth occurrence of character x.

1. "Next" if no minus sign is typed: "previous" if minus sign is typed. The other commands that accept a minus sign are presented in similar fashion.

(continued on next page)

Table A-4 (Cont.)
Alter-Mode Commands

Command	Meaning
L	Print rest of line; recycle to beginning of line with edits incorporated.
P	Print rest of line; recycle to current position with edits incorporated.
Q	Quit Alter mode; restore original line.
[-] [n] Rtext ESC	Replace next/previous n characters with inserted text.
[-] [n] Sx	Skip forward/back to nth occurrence of character x.
[-] [n] W	Skip forward/back n words.
[-] [n] X	Delete next/previous n words.
[-] [n] ^	Invert case of next/previous n characters.
[-] [n] <	Convert next/previous n characters to uppercase.
[-] [n] >	Convert next/previous n characters to lowercase.
[-] [n] BSP	Space back/forward n characters.
RET	Leave Alter mode for this line; print rest of line.
[-] [n] DEL	Space back/forward n characters.
LF	Leave Alter mode for this line; print rest of line.
[-] [n] SP	Space forward/back n characters.
[-] TAB	Skip to end/start of line.
CTRL/R	Retype line to current position.
CTRL/U	Start fresh -- discard edits, recycle to beginning of line, do not print line.

Table A-5
Decide-Mode Commands

Form	Meaning
SP or Y	Yes, make the change as shown.
DEL or N	No, do not make the change.
A	Alter: enter Decide Alter mode with the change already made.
E or Q	End: do not make the change, return to Edit mode.
G	Go: make the change, and make all further changes automatically (leave Decide mode).

Table A-6
SOS Parameters

Name[1]	Set from Edit Mode?	Preset via DCL?[2]	Initial Value	Meaning
BIG	n		–	Displays highest page number
ERROr	n		(null)	Displays last error message
ESCApe	y		(null)	Sets character for input/output of escapes
ID	n		–	Displays current SOS version number
INCRement	y	y	100	Sets increment for numbering inserted lines
ISAVe	y	y	0	Controls auto-W on inserts
LENGth	y		55	Sets page size for List command
LOCAtion	n		^/1	Displays first line in edit buffer
MATCh	y		(null)	Sets special pattern-matching flag character
NAME	n		–	Displays output file specification
PLINes	y	y	16	Sets number of lines printed by P RET
SAVE	y	y	0	Controls auto-W on commands
STARt	y	y	100	Sets starting line number
STEP	y	y	100	Sets increment for initial line numbering
STRIng	n		(null)	Displays current Find and Substitute strings
SUBStitute	n		0	Displays number of matches in last Substitute
.	n		00000/1	Displays current position

[1] Minimum abbreviation is given in uppercase letters.
[2] You can specify these parameters as qualifiers in the initial DCL EDIT command, or by using the symbol-assignment mechanism (see Section 2.9.3).

Table A-7
SOS Switches

Name[1]	Initial Value	Preset via DCL?[2]	Meaning
BAK	on		Tells SOS to create back-up file
DECIde	off	y	Sets auto Decide mode on Substitute
EXACt	off		Requires exact case match for Find, Substitute, content-specification
EXPErt	off	y	Declares experienced SOS user
LINE	on	y	Tells SOS to use existing line numbers
LOWEr	on	y	Tells SOS to accept uppercase, lowercase as is
READonly	off	y	Starts SOS in Read-only mode
SEPArator	off		Tells SOS to treat _$. as alphanumeric
SEQUence	on		Leaves output-file line numbers

[1] Minimum abbreviation is given in uppercase letters.
[2] You can specify these switches as qualifiers in the initial DCL EDIT command, or by using the symbol-assignment mechanism (see Section 2.9.3). Turn off any switch by using the letters NO before its name; thus, NOLINE turns off the LINE switch.

Table A-8
Special Pattern-Matching Constructs

Construct	Internal Representation	Meaning
Find-String Constructs		
?/	CTRL/T	Match any character
?:	|	Match any separator
?<	CTRL/\	Match a space or tab
?%x	CTRL/E	Match any character except x
?)x	CTRL/N	Match 0 or more of the character x
?1x	CTRL/V	Match 1 or more of the character x
?9	CTRL/X	Match any alphanumeric character
?!	CTRL/A	Match any letter (A-Z, a-z)
?&	CTRL/F	Match any uppercase letter (A-Z)
?2	CTRL/W	Match any lowercase letter (a-z)
?+	CTRL/P	Match any decimal digit (0-9)
?>	CTRL/]	Match beginning or end of line
?7c	CTRL/^	Match internal representation of c
Substitute-String Constructs		
?*1?*		The first string matched by a find-string construct
?*2?*		The second string so matched
.		
.		
.		
?*n?*		The nth string so matched
?"		The next string in the sequence matched by a find-string construct

Table A-9
Position Shorthand Characters

Character	Meaning
.	Current line
/.	Current page
^	First line on page
/^	First page in file
*	Last line on page
/*	Last page in file

APPENDIX B

SOS ERROR MESSAGES: NOVICE AND EXPERT FORMS

Table B-1 lists the error messages produced by SOS. In the right-hand column are the forms of the messages that SOS prints if the EXPERT switch is on (see Section 2.9.2). All the messages except one are nonfatal: SOS halts the offending command and returns you to Edit mode. The one fatal error is the last one (You confounded SOS. Sorry.). This error does not occur in normal SOS use. If it does occur, SOS returns you to the monitor. See Section 2.7 for possible recovery procedures in this eventuality.

Table B-1
SOS Error Messages: Novice and Expert Forms

Novice Message	Expert Message
Bad command file name	Name bad
Cannot access HELP file	HELP trouble
Cannot access LIST file	LIST trouble
Cannot delete across pages	Del no pages
Cannot kill initial page mark	K/1 bad
Command invalid for read only file	File RO
Command line I/O error	I/O error
Failed to open command file	Open failure
Illegal file specification	File?
Illegal parameter to switch	Bad param to switch
Illegal search string given	ISS
Illegal specification in line or page number	Bad line/page
Illegal specification in range	Bad range
Illegal switch name	Bad switch
Illegal syntax of command	??
Insufficient line numbers for insertion	No room
Line is too long	Line too long
Line number step too large - lines out of order	Step too big
Lines out of order	Order
Maximum command file depth exceeded	Command files too deep
No such line exists	NSL
Page does not exist	Page?
Range given does not contain any lines	Nothing
Search strings exceeded buffer length	Too long
String not found, search failed	Not there
Too many search strings given	Too many
You confounded SOS. Sorry.	Internal error

APPENDIX C

SLP COMMAND FEATURES

To run SLP, submit a command file containing:

1. Initialization line
2. SLP edit commands
3. DCL $EXIT command

You submit a command file by using either:

- SUBMIT command
- @ (Execute Procedure) command

The general form of the initialization line is:

```
EDIT/SLP[/qualifier] input_file
```

where:

EDIT/SLP	Invokes SLP.
/qualifier	Specifies one or more output qualifiers. This field can be specified using any of the five forms listed in Table C-1.
input_file	Specifies a source file you want to update.

Table C-1
SLP Output Qualifiers

Qualifier	Default	Function
/[NO]AUDIT_TRAIL [:(POSITION:pos,SIZE:len)]	/AUDIT_TRAIL: (POSITION:80,SIZE:8)	Specifies whether an audit trail should be generated, its beginning field and length.
/[NO]LISTING[:file-spec]	/LISTING:file-spec	Specifies whether a listing file should be produced, and lets you rename the listing file.
/[NO]OUTPUT[:file-spec]	/OUTPUT:file-spec	Specifies whether an output file should be produced, and lets you rename the output file.
/[NO]TAB_FILL	/NOTAB_FILL	Inserts tabs at the end of each text line containing an audit trail; /NOTAB_FILL inserts blank spaces.
/[NO]TRUNCATE[:pos]	/NOTRUNCATE	Truncates input lines at the given column position; /NOTRUNCATE disables truncation.

The general form of the SLP edit command is:

```
-locator1[,locator2][,/audittrail/][;comment]
inputline
    .
    .
    .
```

where:

- (minus) Specifies that this is an SLP edit command line.

locator1 Moves current line pointer to specified line. This field can be specified using any of the three syntax forms listed below.

locator2 Represents either an actual line number or the number of lines to be deleted or replaced. This field can be specified using any of the three syntax forms listed below.

/audittrail/ Generates an audit trail which marks new or modified lines in your file. It also lets you change the value of your audit trail.

inputline Inserts new lines into your file immediately following the current line.

;comment An optional comment; SLP ignores any text after a semicolon.

SLP will process a command line with any of the following line locator syntaxes: All forms of the line locator can be used interchangeably in the command line.

1. /stringl[...string2]/
2. / number / [+n]
3. / . /

where:

stringl	Is an ASCII string used to locate the line containing that string. If the locator is specified as /stringl...string2/, SLP locates the line in which the two strings delimit a larger string.
number	Is a sequential number specifying that line in your listing file.
n	Is a decimal value used as an offset from the line specified by the locator.
. (period)	Indicates the current line.

Table C-2
Edit Mode Commands

Command Format	Function
-locator[,/audittrail/] NEW LINE	Add new line(s) of text.
-locator1,locator2[,/audittrail/]	Delete a range of lines.
-locator1,locator2,[,/audittrail/] NEW LINE NEW LINE	Delete a range of lines and replace with new lines.

Table C-3
SLP Operators

Operator	Function
- (minus)	First character of an SLP edit command.
\	Suppresses audit trail generation.
%	Reenables audit trail generation.
@	Invokes a supplemental command file for SLP processing.
/	Terminates SLP edit session and returns to SLP command mode.
<	Allows you to enter characters in an input file that SLP would otherwise interpret as operators.

APPENDIX D

SLP ERROR MESSAGES

Table D-1 lists the error messages produced by SLP.

Table D-1
SLP Error Messages

COMMAND SYNTAX ERROR
ILLEGAL DEVICE NAME
ILLEGAL DIRECTORY
ILLEGAL ERROR/SEVERITY CODE p1 p2 p3
ILLEGAL FILE NAME
ILLEGAL GET COMMAND LINE ERROR
ILLEGAL SWITCH
INDIRECT COMMAND SYNTAX ERROR
INDIRECT FILE DEPTH EXCEEDED
INDIRECT FILE OPEN FAILURE
I/O ERROR COMMAND INPUT FILE
I/O ERROR COMMAND OUTPUT FILE
I/O ERROR CORRECTION INPUT FILE
I/O ERROR LINE LISTING FILE
I/O ERROR SOURCE OUTPUT FILE
LINE NUMBER ERROR
OPEN FAILURE CORRECTION INPUT FILE
OPEN FAILURE SOURCE OUTPUT FILE
PREMATURE EOF COMMAND INPUT FILE
PREMATURE EOF CORRECTION INPUT FILE

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

W

VAX-11
Text Editing
Reference Manual
AA-D029A-TE

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

_____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Please cut along this line.

Do Not Tear - Fold Here and Tape

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

Do Not Tear - Fold Here