
Open VMS SUMSLP Utility Manual

Order Number: AA-PS6EA-TE

May 1993

SUMSLP is a batch-oriented editor that allows multiple update files to be applied to a single input file.

Revision/Update Information: This document supersedes the *VMS SUMSLP Utility Manual*, Version 5.2.

Software Version: OpenVMS AXP Version 1.5
OpenVMS VAX Version 6.0

**Digital Equipment Corporation
Maynard, Massachusetts**

May 1993

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

© Digital Equipment Corporation 1993.

All Rights Reserved.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: AXP, DECwindows, Digital, OpenVMS, VAX, VAX DOCUMENT, VMS, and the DIGITAL logo.

The following is a third-party trademark:

PostScript is a registered trademark of Adobe Systems, Inc.

ZK4559

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface	v
SUMSLP Description	SUM-1
1 SUMSLP Utility	SUM-1
2 SUMSLP Files	SUM-1
2.1 The Input Source File	SUM-1
2.2 The SUMSLP Update File	SUM-1
2.3 The Output File	SUM-3
2.4 The Listing File	SUM-3
3 Specifying SUMSLP Editing Commands	SUM-3
3.1 SUMSLP Operators	SUM-3
3.2 Editing Command Format	SUM-4
3.2.1 Examples of Editing Commands	SUM-5
3.3 How SUMSLP Processes Files	SUM-6
3.4 Adding Lines to a File	SUM-7
3.5 Deleting Lines from a File	SUM-9
3.6 Replacing Lines in a File	SUM-10
3.7 Changing the Audit Trail Text	SUM-11
4 Running SUMSLP from a Command File	SUM-11
5 SUMSLP Messages	SUM-12
SUMSLP Usage Summary	SUM-13
SUMSLP Qualifiers	SUM-14
/LISTING	SUM-15
/OUTPUT	SUM-16
/HEADER	SUM-17
SUMSLP Positional Qualifier	SUM-18
/UPDATE	SUM-19
SUMSLP Examples	SUM-20
Index	
Figures	
SUM-1 Files Used During SUMSLP Processing	SUM-2

Tables

SUM-1 SUMSLP Operators SUM-3

Intended Audience

This manual is intended for all users familiar with VMS system concepts who want to modify or update input source files.

Document Structure

This document consists of the following four sections:

- Description—Provides a full description of the SUMSLP Utility (SUMSLP).
- Usage Summary—Outlines the following SUMSLP information:
 - Invoking the utility
 - Directing output
- Qualifiers—Describes SUMSLP qualifiers and positional qualifier, including format, parameters, and examples.
- Examples—Provides additional SUMSLP examples.

Associated Documents

The following manuals provide additional information:

- *OpenVMS User's Manual*
- *OpenVMS DCL Dictionary*
- *OpenVMS System Messages and Recovery Procedures Reference Manual*
- *OpenVMS System Manager's Manual*

Conventions

In this manual, every use of VMS means both the OpenVMS AXP and the OpenVMS VAX operating system.

The following conventions are used in this manual:

Ctrl/ <i>x</i>	A sequence such as Ctrl/ <i>x</i> indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 <i>x</i>	A sequence such as PF1 <i>x</i> indicates that you must first press and release the key labeled PF1, then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
...	A horizontal ellipsis in examples indicates one of the following possibilities: <ul style="list-style-type: none">• Additional optional arguments in a statement have been omitted.• The preceding item or items can be repeated one or more times.• Additional parameters, values, or other information can be entered.
.	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
()	In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[]	In format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in a VMS file specification, or in the syntax of a substring specification in an assignment statement.)
{ }	In format descriptions, braces surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason. Boldface text is also used to show user input in online versions of the manual.
<i>italic text</i>	Italic text emphasizes important information, indicates variables, and indicates complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error <i>number</i>), command lines (for example, /PRODUCER= <i>name</i>), and command parameters in text.
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.

numbers

All numbers in text are assumed to be decimal, unless otherwise noted. Nondecimal radices—binary, octal, or hexadecimal—are explicitly indicated.

SUMSLP Description

This section describes the use of the SUMSLP utility. The required files are listed together with basic examples of their use. Also included are examples of the editing format and the edited output.

1 SUMSLP Utility

The SUMSLP Utility (SUMSLP) is a batch-oriented editor that allows you to update source files. You enter text changes and editing commands in update files. SUMSLP merges the update files containing the changes with the source file that you specify and gives you a record of the changes in the form of an audit trail.

SUMSLP produces a new copy of the input source file containing the additions and changes that you specified in the SUMSLP update files.

2 SUMSLP Files

SUMSLP requires two types of input files: an input source file and a SUMSLP update file. These files are described in Section 2.1 and Section 2.2.

SUMSLP produces an output file, described in Section 2.3, which is the permanently updated copy of the input file. It includes the changes SUMSLP makes to the input file.

You can generate a listing file, which contains the original lines, the inserted lines, and an audit trail. See Section 2.4 for a complete description.

Figure SUM-1 shows the relationships of the SUMSLP input and output files. The files are described in the following sections.

2.1 The Input Source File

The input source file is the file you want to update. The source file must contain no more than 32,767 lines.

SUMSLP editing commands require line numbers to locate where your source file should be changed. Use the `/LISTING` qualifier, which gives a numbered listing of your source file, to determine the line numbers that you want to update. See the Qualifiers Section for more information about `/LISTING`.

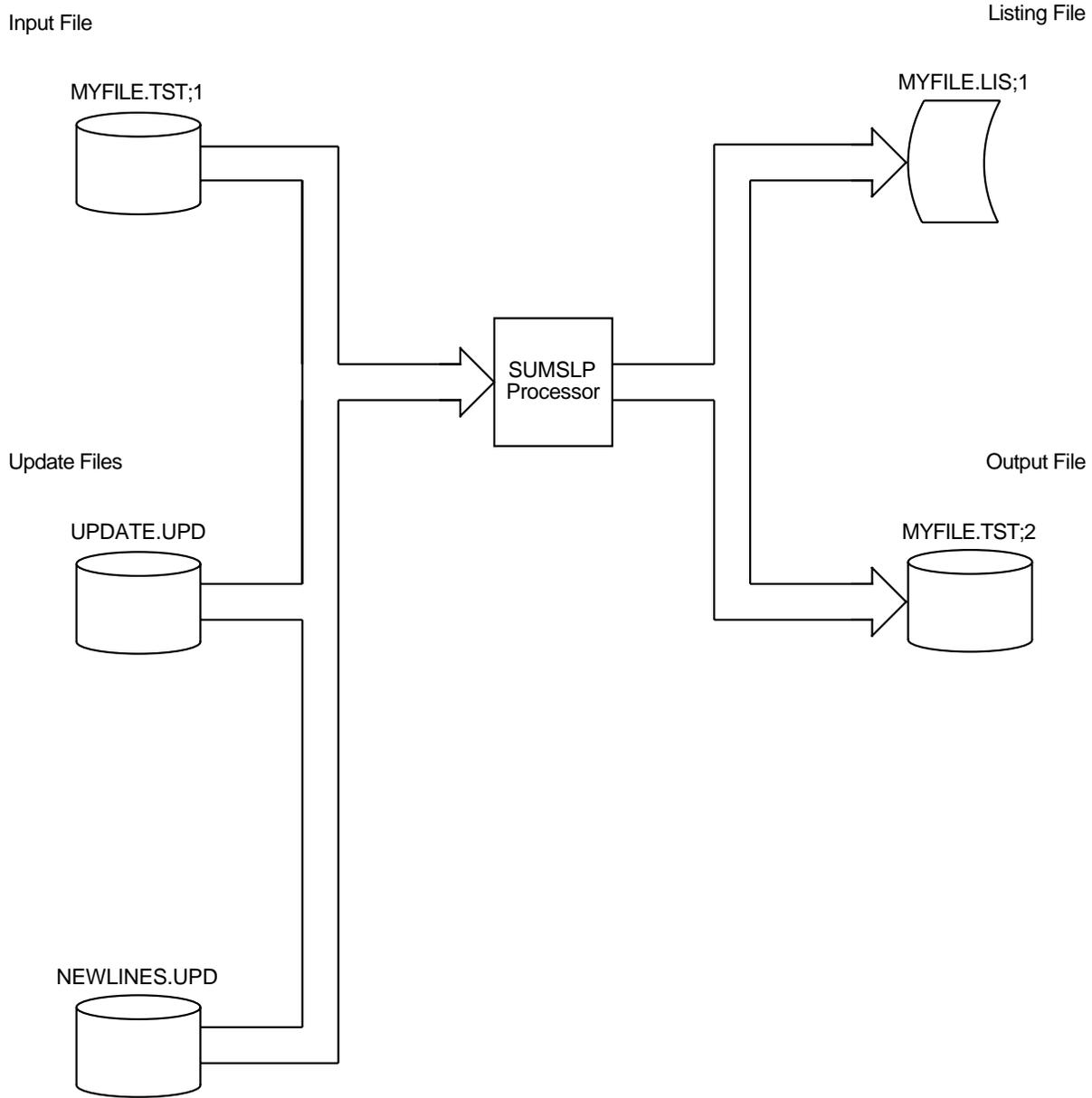
2.2 The SUMSLP Update File

The SUMSLP update file contains input lines with the changes you want to make, and edit command lines that perform the update procedure. You can create as many update files as you want. The following steps describe how to create an update file:

1. Using an interactive text editor, create an update file that contains the following elements:
 - a. SUMSLP editing commands that list changes to the input file.
 - b. Input lines of text to be added to the source file, either as additional lines or to replace old lines. Use SUMSLP operators, described in Section 3.1, to perform these functions.

SUMSLP Description

Figure SUM-1 Files Used During SUMSLP Processing



ZK-0865-GE

- c. The SUMSLP terminator, a single slash (/) in column 1, which causes SUMSLP to begin its processing (updating) of the file.
2. Enter the EDIT/SUM command, as shown below, to invoke SUMSLP and to indicate which files you want to update:

```
$ EDIT/SUM input-file/UPDATE=update-file-spec
```

This command line is described in the Format section.

Note

SUMSLP has a line limit of 32,767 lines. Do not use large SUMSLP update files produced by the DIFFERENCES/SLP DCL command.

2.3 The Output File

The SUMSLP output file contains the input source file updated by the additions and changes specified in the SUMSLP command files. It does not include an audit trail or line numbers.

If you do not include a file specification for the output file with the /OUTPUT qualifier in the EDIT/SUM command, the output file takes the same file name as the input source file, with a new version number.

2.4 The Listing File

The SUMSLP listing file, which displays original lines, inserted lines, and an audit trail, is produced if you specify the /LISTING qualifier in the EDIT/SUM command. If you do not specify another name for it, it takes the same file name as the input source file, with the default file type of LIS.

3 Specifying SUMSLP Editing Commands

Use SUMSLP editing commands to update source files by adding, deleting, and replacing lines in a file. These commands include characters that SUMSLP interprets as operators. This section describes these operators and the general form for specifying SUMSLP editing commands. It then describes specific editing commands.

3.1 SUMSLP Operators

When SUMSLP encounters any of the characters listed in Table SUM-1 as the first character in an input line, it interprets the character as an operator.

Table SUM-1 SUMSLP Operators

Operator	Meaning
- (minus sign)	Must appear as the first character of a SUMSLP editing command
\ (backslash)	Suppresses audit trail generation
% (percent sign)	Enables audit trail generation
/ (slash)	Terminates the editing session
< (less-than character)	Escape character

The percent sign (%) operator enables audit trail generation after generation has been suppressed by the backslash (\) operator.

The less-than (<) operator enables the use of SUMSLP operators (including <) as standard characters. For example, if you enter </ (in columns 1 and 2), the slash is interpreted as a character, and your editing session is not terminated.

SUMSLP Description

3.2 Editing Command Format

The format of a SUMSLP editing command is as follows:

```
-locator1,locator2,/audittrail/;comment  
inputline  
.  
.  
.
```

All fields in the command line are position-dependent; commas must be included to separate fields. Commas must also be included for omitted fields.

Command Parameters

- (minus sign)

A minus-sign operator indicates that this is a SUMSLP editing command line.

locator1

A line locator that causes SUMSLP to move the current line pointer to a line you specify. If you specify only one locator (locator1), the current line pointer is moved to that line, and SUMSLP reads the next line in the editing command file. Used with locator2, locator1 defines the beginning of a range of lines.

locator2

A line locator that defines the end of a range of lines, beginning with locator1, to be deleted or replaced. Locator1 and locator2 fields can be specified using either

of the following locator forms: $\left[\begin{array}{c} \text{number} \\ \cdot \\ \end{array} \right]$

number

A number in the range of 1 through 32767 to which the current line pointer is moved.

. (period)

A period represents the current line. The following example shows how to delete a single line (line 2) using the period locator:

```
-2,.  
/
```

Both forms of the line locator can be specified in a command line.

SUMSLP edits files sequentially. Once the line pointer moves past a given line in the file, it cannot be returned to that line.

/audittrail/

A character string that marks new or replaced lines in the file. This argument must be delimited by slashes (/). If you do not include two locator fields in the editing command, precede the audit trail specification with two commas, as shown in the following example:

```
-5,,/**08-JUN**/
```

Usually the first character of the audit trail is set to match the comment delimiter of the source file being edited (for example, a semicolon (;) as shown above). Thus, the audit trail string is ignored when the source file is compiled or assembled because the audit trail begins with the comment delimiter.

An audit trail is produced automatically until either changed or suppressed by omitting the audit trail string, as shown in the following examples:

```
-5,,
/

-5
/
```

The default audit trail for each line added is **;**NEW****.

To indicate the total number of lines deleted or replaced, the listing file contains a marker. The marker is placed on the first unchanged line following a deletion or replacement and has the form **-n**, where **n** is the number of lines deleted or replaced. In the following example, **-5** indicates that 5 lines were deleted:

```

.
.
.
4 KLM
-5          10 987
```

;comment

An optional comment. SUMSLP ignores any text after a semicolon.

inputline

A line of new text that you enter, which is inserted into the file immediately following the current line. You can enter any number of input lines.

3.2.1 Examples of Editing Commands

The following example shows the contents of a SUMSLP update file named **MYFILE.UPD**. The file contents are followed by the command line that updates the input source file named **MYFILE.TST**. (Section 2.2 describes update files.)

MYFILE.UPD contents:

```
-3                                     1
INSERT THIS LINE AFTER LINE 3         2
-4,4                                   1
DELETE LINE 4 AND REPLACE IT WITH THIS LINE 2
/                                       3

$ EDIT/SUM MYFILE.TST/UPDATE=MYFILE.UPD 4
```

- 1 Editing command
- 2 Input line
- 3 Terminator
- 4 Command line

The next example illustrates the generation of a listing file. **MYFILE.TST** (input source file) contents:

```
ONE
TWO
THREE
FOUR
FIVE
SIX
```

SUMSLP Description

SEVEN
EIGHT
NINE
TEN

In this example, two update files have been created. The first, UPDATE.UPD, contains the following editing commands and text:

```
-3,3,/:21-MAR/  
REPLACED LINE  
/
```

The second update file, NEWLINES.UPD, contains the following editing commands and text:

```
-7,,/:22-MAR/  
NEW LINE  
/
```

When the commands in these SUMSLP update files are applied to the input source file, SUMSLP produces the listing file MYFILE.LIS.

The SUMSLP command for this example contains the following file specification and qualifiers:

```
$ EDIT/SUM MYFILE.TST/LISTING/UPDATE=(UPDATE.UPD,NEWLINES.UPD)
```

The listing file, MYFILE.LIS, now contains the following lines:

```
1 ONE  
2 TWO  
;21-MAR .1 REPLACED LINE  
-1 4 FOUR  
5 FIVE  
6 SIX  
7 SEVEN  
;22-MAR .1 NEW LINE  
8 EIGHT  
9 NINE  
10 TEN
```

3.3 How SUMSLP Processes Files

SUMSLP applies the edits that you specify in the SUMSLP update files to the source lines of the input source file. When a list of update files is specified with the /UPDATE qualifier, the editing commands in the various files are arranged according to the following rules:

1. Using the line number pointed to by locator1, SUMSLP applies editing commands sequentially, in ascending order. All edits that do not overlap or conflict with any other edits are applied to the source file without any further processing.
2. Conflicting editing commands are resolved according to the precedence of the SUMSLP command file in which the commands occur. Precedence is determined by the position of the file specifications following /UPDATE. The file specification listed last has the highest precedence. In the following example, the edit commands in the CHANGE02.UPD file would have the highest precedence:

```
$ EDIT/SUM MYFILE.TST/UPDATE=(CHANGE01.UPD,CHANGE02.UPD)
```

All inserts to the same source line are included in the output file; those from the SUMSLP update file with the highest precedence appear first.

An operation that deletes or replaces a line affects the specified line and any lower precedence inserts or replacements to the same line. A deletion that specifies a range of lines deletes all lines occurring in that range, including inserted lines from SUMSLP update files of lower precedence.

3.4 Adding Lines to a File

The SUMSLP editing command for adding lines to a file, as shown below, contains only one locator field:

```
-locator1,,/audittrail;/comment
```

If, in the locator1 field, you specify a line number, SUMSLP inserts new lines after that line number. The new lines, which are taken from the update file, are those text lines that follow the SUMSLP command up to, but not including, the next SUMSLP command line.

Because there is only one locator field, the audit trail specification must be preceded by two commas.

In a SUMSLP listing, a period precedes the line numbers of inserted lines to distinguish them from original lines. Inserted line numbers begin with .1 at the start of each series of new lines, as shown in the following excerpt from a listing file.

```

.
.
.
3 THREE
;21-MAR      .1 INSERTED LINE
4 FOUR
.
.
.
```

The following example shows how lines are added to a file. The input source file (MYFILE.TST) consists of the following ten lines:

```

1  ABC
2  DEF
3  GHI
4  KLM
5  123456789
6  456
7  789
8  CBA
9  XYX
10 987
```

SUMSLP Description

The SUMSLP update file (UPDATE.UPD) consists of the following editing commands and lines of text:

```
-5, ,/; **08-JUN**/  
INSERT THIS LINE AFTER LINE 5  
/
```

The SUMSLP command line contains the following file specification and qualifiers:

```
$ EDIT/SUM MYFILE.TST/OUTPUT/LISTING=NEWFILE.LIS/UPDATE=UPDATE.UPD
```

The input file is MYFILE.TST. The /OUTPUT qualifier generates an output file with the same name and a higher version number (MYFILE.TST;2). The /LISTING qualifier generates the listing file NEWFILE.LIS;1. The /UPDATE qualifier specifies the file UPDATE.UPD as the update file. NEWFILE.LIS;1 contains the following lines:

```
1 ABC  
2 DEF  
3 GHI  
4 KLM  
5 123456789  
; **08-JUN** .1 INSERT THIS LINE AFTER LINE 5  
6 456  
7 789  
8 CBA  
9 XYX  
10 987
```

In the preceding listing example, SUMSLP has applied line numbers and added an audit trail (; **08-JUN**) to the line following line 5.

The next example uses the updated input source file (MYFILE.TST;2) and a new SUMSLP update file (NEWTEXT.UPD) containing the following lines:

```
-4, ,/; **09-JUN**/  
THIS IS NEW TEXT  
/
```

The command line contains the following file specification and qualifiers:

```
$ EDIT/SUM MYFILE.TST/OUTPUT/LISTING=NEWFILE.LIS/UPDATE=NEWTEXT.UPD
```

SUMSLP processing generates the output file MYFILE.TST;3 and the listing file NEWFILE.LIS;2. NEWFILE.LIS;2 contains the following lines:

```
1 ABC
2 DEF
3 GHI
4 KLM
; **09-JUN** .1 THIS IS NEW TEXT
5 123456789
6 INSERT THIS LINE AFTER LINE 5
7 456
8 789
9 CBA
10 XYX
11 987
```

SUMSLP has numbered the lines in sequence. In this example, the new input line is inserted after line 4.

3.5 Deleting Lines from a File

The following SUMSLP editing command for deleting lines from a file contains two locator fields:

```
-locator1,locator2,/audittrail/;comment
```

The first field, locator1, specifies the line where SUMSLP is to begin deleting lines; locator2 specifies the last line to be deleted. SUMSLP deletes all lines from locator1 through locator2, inclusive.

The following example shows how to delete lines from a file using SUMSLP. The input source file (TESTFILE.TST) consists of the following ten lines:

```
1 ABC
2 DEF
3 GHI
4 KLM
5 123456789
6 456
7 789
8 CBA
9 XYX
10 987
```

The SUMSLP update file (CHANGE01.UPD) contains the following editing commands:

```
-5,9
/
```

The SUMSLP command line contains the following file specification and qualifiers:

```
$ EDIT/SUM TESTFILE.TST/OUTPUT/LISTING=ALPHA.LIS/UPDATE=CHANGE01.UPD
```

SUMSLP Description

SUMSLP processing creates the files TESTFILE.TST;2 and ALPHA.LIS;1. ALPHA.LIS;1 contains the following lines:

```
1 ABC
2 DEF
3 GHI
4 KLM
-5      10 987
```

In this example, SUMSLP first deletes line 5, which is specified by the locator1 entry. It then continues to delete lines until it deletes line 9, which is specified by the locator2 entry.

Using the original input source file from the example above, the next example shows how to delete a single line using the period locator. The new update file (CHANGE02.UPD) contains the following lines:

```
-2, .
/
```

The SUMSLP command line contains the following file specification and qualifiers:

```
$ EDIT/SUM TESTFILE.TST/OUTPUT/LISTING/UPDATE=CHANGE02.UPD
```

SUMSLP processing creates the files TESTFILE.TST;2 and TESTFILE.LIS;1. TESTFILE.LIS;1 contains the following lines:

```
1 ABC
-1      3 GHI
4 KLM
5 123456789
6 456
7 789
8 CBA
9 XYX
10 987
```

SUMSLP moves the current line pointer to line 2 and then finds the period as the second locator field. Since the second locator field is specified, SUMSLP interprets the editing command as a delete operation and deletes the line containing DEF.

3.6 Replacing Lines in a File

A replacement is a deletion followed by the insertion of new text. The number of lines deleted does not need to match the number of lines added. To replace lines in a file, use the two-locator command form. The first line locator field specifies the first line to be deleted. The second line locator field specifies the last line in the range to be deleted, which, for replacement operations, is the line where new text is to be inserted.

The example below shows how to delete lines from a file and replace them with new lines. The input source file (SOURCE.TST) consists of the following seven lines:

```

1   ABC
2   DEF
3   GHI
4   123456789
5   BCN
6   CRB
7   BUR

```

The SUMSLP update file (REPLACE.UPD) contains the following editing command and text lines:

```

-2,3,/,**LINE*REPLACED**/
XXXXXX
YYYYYY
ZZZZZZ
/

```

The SUMSLP command line contains the following file specification and qualifiers:

```
$ EDIT/SUM SOURCE.TST/LISTING/OUTPUT/UPDATE=REPLACE.UPD
```

SUMSLP processing generates the files SOURCE.TST;2 and SOURCE.LIS;1. SOURCE.LIS contains the following lines:

```

1 ABC
**LINE*REPLACED** .1 XXXXXX
**LINE*REPLACED** .2 YYYYYY
**LINE*REPLACED** .3 ZZZZZZ
-2 4 123456789
5 BCN
6 CRB
7 BUR

```

3.7 Changing the Audit Trail Text

The following SUMSLP edit command changes the text of the audit trail:

```
-,./newtrail/
```

Here, newtrail is the new text of the audit trail. All subsequent lines added will include the new audit trail text.

When you create a new audit trail, you may want to set the first character of the string to correspond to the source files's comment delimiter so that the audit trail string is ignored when the source file is compiled or assembled.

4 Running SUMSLP from a Command File

If you find that you are often using the same command line and editing commands, you might want to create a command file that contains all the commands. To run SUMSLP from a command file, create a file that contains the required SUMSLP editing commands, input lines, and the SUMSLP terminator (/) in the first column of the last line.

SUMSLP Description

The following is an example of a command file named UPDATE.COM:

```
$ EDIT/SUM MYFILE.TST/UPDATE=SYSS$INPUT
-3
INSERT THIS LINE AFTER LINE 3
-4,4
DELETE LINE 4 AND REPLACE IT WITH THIS LINE
/
```

Execute UPDATE.COM by using the Execute Procedure (@) command, as follows:

```
$ @UPDATE
```

Because the file type is the default file type COM, omit it from the DCL command line. (See the *OpenVMS User's Manual* handbook for information on using command procedures and running batch jobs.) When SUMSLP finishes its processing, the DCL prompt is issued.

5 SUMSLP Messages

The *OpenVMS System Messages and Recovery Procedures Reference Manual* lists the diagnostic messages issued by SUMSLP and provides explanations and suggested user response to these messages.

SUMSLP Usage Summary

The SUMSLP Utility is a batch-oriented editor that allows you to update source files. You enter text changes and editing commands in update files. SUMSLP merges the update files containing the changes with the source file that you specify and gives you a record of the changes in the form of an audit trail. SUMSLP produces a new copy of the input source file containing the additions and changes that you specified in the SUMSLP update files.

Format

```
EDIT/SUM input-file
```

Command Parameter

input-file

The file specification for the source file to be edited.

Usage Summary

SUMSLP can be run either indirectly from a command procedure or interactively from your terminal. To invoke SUMSLP interactively, enter the following DCL command:

```
EDIT/SUM input-file
```

To invoke SUMSLP from a command procedure, precede the command with a dollar sign (\$).

You can update source files using SUMSLP editing commands, which allow you to add, delete, or replace lines in a file (see Section 3). There are no privileges or restrictions in running SUMSLP.

SUMSLP Qualifiers

SUMSLP qualifiers control the generation and format of the listing file and the output file. Use them to control the output options associated with these files. The following sections illustrate the use of SUMSLP qualifiers.

/LISTING

Controls whether a sequence-numbered listing file showing the original and inserted lines and an audit trail is produced during the editing process.

Format

`/[NO]LISTING[=file-spec]`

Parameters

file-spec

The file specification of the listing file.

Description

If you do not include a file specification, the listing file takes the same name as the input file with a default file type of LIS. The default is `/NOLISTING`. The listing file is described in Section 2.4.

`/LISTING` is a command qualifier only.

SUMSLP /OUTPUT

/OUTPUT

Specifies the output file to be created in the editing operation.

Format

/[NO]OUTPUT[=file-spec]

Parameters

file-spec

The file specification of the output file.

Description

If you do not specify /OUTPUT or omit the file specification, the output file has the same name and type as the input file with a new version number. The output file is described in Section 2.3.

/OUTPUT is a command qualifier only.

/HEADER

Controls whether the output file is created as a sequential file of variable length records in fixed-length control (VFC) format with the line insert number and audit trail information in the record header block (RHB).

Format

/[NO]HEADER

Parameters

None.

Description

The default is /NOHEADER. Because use of the /HEADER qualifier affects the format of the output file, the /OUTPUT qualifier should be specified with it.

SUMSLP Positional Qualifier

The SUMSLP positional qualifier /UPDATE controls the selection of the update file or files. The following section describes the use of /UPDATE.

/UPDATE

Indicates the file or files containing the editing commands and changes to be applied to the input source file.

Format

/UPDATE[=(update-file-spec[,...])]

Parameters

update-file-spec[,...]

The file or files containing the editing commands and changes to be applied to the input source file.

Description

If multiple file specifications are listed, separate them with commas, and enclose the list in parentheses. Section 3.3 describes SUMSLP processing of multiple update files. The default file type of these files is UPD. Default values for the other elements of the file specification are initially taken from the input file specification; after the first file specification in a list, values default to those of the immediately preceding file specification.

If no file specification or list of file specifications is given, SUMSLP attempts to open a single update file with the same file name as the input file and a file type of UPD.

If you do not include the /UPDATE qualifier in the command line, SUMSLP does not search for an update file. It generates the specified output or listing file with no changes applied. Enter the EDIT/SUM command with the /LISTING qualifier, but without the /UPDATE qualifier, to generate a numbered listing of your source program.

Note

SUMSLP has a line limit of 32,767 lines. Do not use large SUMSLP update files produced by the DIFFERENCES/SLP DCL command.

/UPDATE is a file qualifier only.

SUMSLP Examples

1. \$ EDIT/SUM FILE1.MAR/UPDATE

In Example 1, the input source file FILE1.MAR is updated with the SUMSLP update file FILE1.UPD.

2. \$ EDIT/SUM FILE2.MAR/UPDATE=UPD2

In Example 2, the input source file FILE2.MAR is updated with the SUMSLP update file UPD2.UPD because UPD is the default file type.

3. \$ EDIT/SUM FILE3.MAR/UPDATE=(UPD3A,UPD3B.ENH,UPD3C)

In Example 3, the input source file FILE3.MAR is updated with the merged contents of SUMSLP update files UPD3A.UPD, UPD3B.ENH, and UPD3C.ENH.

Index

A

Audit trail

- changing the value of, SUM-11
- listing, SUM-15

C

Command file

- running SUMSLP from a, SUM-11

D

DIFFERENCES/SLP DCL command, SUM-3

E

Editing commands

- format of, SUM-4
- how to add lines, SUM-7, SUM-8
- how to change audit trail text, SUM-11
- how to delete lines, SUM-9, SUM-10
- how to replace lines, SUM-10
- how to specify, SUM-3
- using command parameters, SUM-4
- using locator field parameters, SUM-4
- using operators, SUM-3

EDIT/SUM DCL command, SUM-2, SUM-13

Examples, SUM-20

- adding lines, SUM-7, SUM-8
- audit trail text, SUM-11
- deleting lines, SUM-9
- listing file, SUM-6

F

File

- adding lines to a, SUM-7
- input source, SUM-1
- listing, SUM-3, SUM-6, SUM-15
- output, SUM-3, SUM-16
 - format, SUM-17
- update, SUM-1, SUM-19
 - produced by DIFFERENCES/SLP DCL command, SUM-3
 - restrictions, SUM-19

File processing, SUM-6

H

/HEADER qualifier, SUM-17

I

Input source file, SUM-1, SUM-19

L

/LISTING qualifier, SUM-15

M

Messages, SUM-12

O

Operators, SUM-3

Output file, SUM-3, SUM-16, SUM-17

/OUTPUT qualifier, SUM-16

Q

Qualifiers, SUM-14 to SUM-19

S

SUMSLP Utility

- command, SUM-13
- directing output, SUM-13
- input source file, SUM-1
- invoking, SUM-2, SUM-13
- output file, SUM-3, SUM-16, SUM-17
- qualifiers, SUM-14 to SUM-19
- update file, SUM-1, SUM-19

U

Update file, SUM-1, SUM-19

/UPDATE qualifier, SUM-19

