# VAX/VMS SUMSLP Utility Reference Manual

Order Number: AA–Z432A–TE

**September 1984**

SUMSLP is a batch-oriented editor that allows multiple update files to be applied to a single input file. The update files are combined according to fixed rules.

**Revision/Update Information:** This is a new manual.

**Software Version:** VAX/VMS Version 4.0

**digital equipment corporation**
**maynard, massachusetts**

**September 1984**



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

The following are trademarks of Digital Equipment Corporation:

| | | |
|---|---|---|
| DEC | DIBOL | UNIBUS |
| DEC/CMS | EduSystem | VAX |
| DEC/MMS | IAS | VAXcluster |
| DECnet | MASSBUS | VMS |
| DECsystem–10 | PDP | VT |
| DECSYSTEM–20 | PDT | |
| DECUS | RSTS | |
| DECwriter | RSX | digital |

ZK–2305

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a registered trademark of the American Mathematical Society.

# SUMSLP Contents

# Preface

## Intended Audience

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

## Structure of This Document

This document is composed of four major sections.

The Format Section is an overview of the SUMSLP Utility and is intended as a quick reference guide. The format summary contains the DCL command that invokes the SUMSLP Utility, listing all command and positional qualifiers and parameters. The usage summary describes how to invoke and exit from the SUMSLP Utility, how to direct output, and any restrictions that you should be aware of.

The description summary explains how to use the SUMSLP Utility.

The Qualifier Section describes each DCL command qualifier. Qualifiers appear in alphabetical order.

The Examples Section contains examples of common operations that you perform with the SUMSLP Utility.

## Associated Documents

Three other manuals that may be of use:

- *Guide to Using DCL and Command Procedures on VAX/VMS*
- *VAX/VMS DCL Dictionary*
- *Guide to VAX/VMS System Management and Daily Operations*

## Conventions Used in This Document

| Convention | Meaning |
|---|---|
| RET | A symbol with a one- to three-character abbreviation indicates that you press a key on the terminal, for example, RET . |
| CTRL/x | The phrase CTRL/x indicates that you must press the key labeled CTRL whil you simultaneously press another key, for example, CTRL/C, CTRL/Y, CTRL/O. |

| Convention | Meaning |
|---|---|
| $ SHOW TIME<br>05-JUN-1982 11:55:22 | Command examples show all output lines or prompting chracters that the system prints or displays in black letters. All user-entered commands are shown in red letters. |
| $ TYPE MYFILE.DAT<br>.<br>.<br>. | Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown. Vertical ellipsis in coding examples indicate that lines of code not pertinent to the example are omitted. |
| file-spec,... | Horizontal ellipsis indicates that additional parameters, values, or information can be entered. |
| [logical-name] | Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.) |
| quotation marks<br>apostrophes | The term quotation marks is used to refer to double quotation marks (″). The term apostrophe (′) is used to refer to a single quotation mark. |

# New and Changed Features

This manual documents the SUMSLP Utility in Version 4.0 of VAX/VMS. This section summarizes the main technical changes from Version 3.0:

- The SLP editor is no longer supported.
- The /HEADER command qualifier has been included in the list of SUMSLP command qualifiers. This qualifier controls the structure of the output file.

# SUMSLP

SUMSLP is a batch-oriented editor that allows multiple update files to be applied to a single input file. The update files are combined according to fixed rules.

## FORMAT

**EDIT/SUM** *input-file*

| Command Qualifiers | Defaults |
|---|---|
| */HEADER/OUTPUT=file-spec* | *None.* |
| */LISTING[=file-spec]* | *See text.* |
| */OUTPUT[=file-spec]* | *See text.* |

| Positional Qualifier | Default |
|---|---|
| */UPDATE[=(update-file-spec[,...])]* | *See text.* |

**Command Parameter**

*input-file*

The file specification for the source file to be edited (see Section 1.1).

## usage summary

**Invoking**

SUMSLP can be run either indirectly from a command procedure or interactively from your terminal. To invoke SUMSLP interactively, you issue a command line as shown above in response to the DCL prompt. To invoke SUMSLP from a command procedure, precede the command with a dollar sign ($).

**Directing Output**

You can update source files through the use of SUMSLP editing commands, which allow you to add, delete, or replace lines in a file (see Section 3).

**Privileges/Restrictions**

None.

## commands

**Syntax**

-locator1[,locator2][,/audittrail/][;comment]

**SUMSLP Commands**

None.

**SUMSLP Operators:**

| Operator | Meaning |
|---|---|
| - (minus sign) | First character of a SUMSLP editing command |
| \ (backslash) | Suppresses audit trail generation |
| % (percent sign) | Reenables audit trail generation |

| Operator | Meaning |
|---|---|
| / (slash) | Terminates the editing session |
| < (less-than character) | Escape character |

## DESCRIPTION

Input to SUMSLP consists of an input source file that you want updated and one or more update files containing text lines and edit command lines that specify the update operations to be performed. SUMSLP locates lines to be changed by means of "locators" (line numbers).

SUMSLP output is an updated copy of the input source file. It contains the additions and changes specified in the SUMSLP command file(s).

## 1 SUMSLP Files

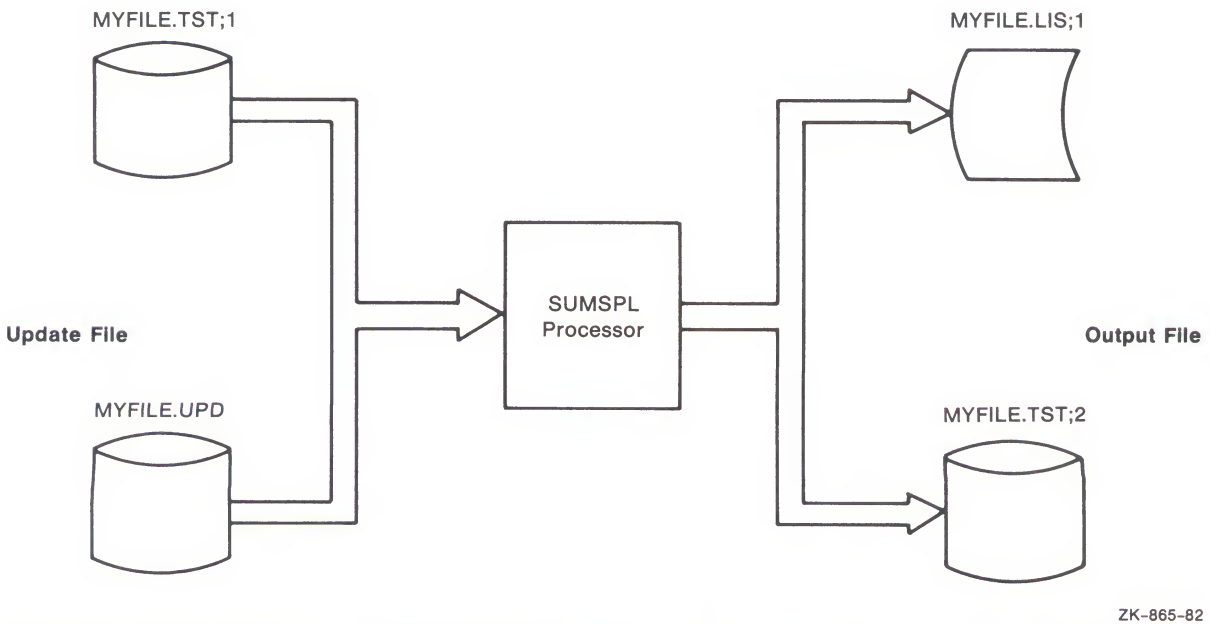
SUMSLP requires two types of input files: an input source file and a SUMSLP update file. These files are described in Sections 1.1 and 1.2.

The output file, described in Section 1.3, is the permanently updated copy of the input file. It shows the changes SUMSLP makes to the input file.

You can also generate a listing file, as described in Section 1.4.

Figure SUM–1 shows the relationships among the SUMSLP output and input files. The contents of the various files in this figure are described in the following sections.

**Figure SUM–1 Files Used During SUMSLP Processing**



## 1.1 The Input Source File

The input source file is the file to be updated by SUMSLP. It can contain any number of lines.

To use SUMSLP effectively, you should obtain a sequence-numbered listing of the input file from which you can determine what editing commands you will issue. Section 1.4 describes how to generate such a listing using the /LISTING qualifier. However, the input source file actually updated by SUMSLP can have any kind of line numbers.

## 1.2 The SUMSLP Update File

The SUMSLP update file contains command lines and input lines used to alter the input file. To update the input file you must perform the following two-step procedure:

1 Create an update file that contains the following elements:

   a SUMSLP editing command lines that define changes to the input file (see Section 3).

   b Input lines, that is, lines of text that are to be inserted into the output file, either as new lines or to replace old lines.

   c The SUMSLP terminator, a single slash ( / ) in column 1, that causes SUMSLP to begin its processing (updating) of the file.

2 Enter the EDIT/SUM command to invoke SUMSLP and specify what files to process:

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

   This command line is described in the Format Section.

An interactive text editor is usually used to create SUMSLP update files. Once you have created the file, you can perform the update by entering the appropriate EDIT/SUM command line.

The example below shows the contents of a SUMSLP update file named MYFILE.UPD. The file is followed by the command line that will update the input source file named MYFILE.TST. The numbers to the right of the example corresponds to the elements listed above.

```
MYFILE.UPD contents:
     -3                                          (1a)
     INSERT THIS LINE AFTER LINE 3               (1b)
     -4,4                                        (1a)
     DELETE LINE 4 AND REPLACE IT WITH THIS LINE (1b)
     /                                           (1c)
$ EDIT/SUM MYFILE.TST/UPDATE=MYFILE.UPD          (2)
```

## 1.3 The Output File

The SUMSLP output file contains the input source file as updated by the additions and changes specified in the SUMSLP command file(s). 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 version number one higher than the existing version number.

## 1.4 The Listing File

The SUMSLP listing file 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 file type of LIS. You can specify another file type, but LIS is the default.

The following example illustrates the generation of a listing file. The input source file, named MYFILE.TST, looks like this:

```
ONE
TWO
THREE
FOUR
FIVE
SIX
SEVEN
EIGHT
NINE
TEN
```

There are two SUMSLP update files. The first, UPDATE.UPD, contains the following editing commands:

```
-3,3,/;21-MAR/
INSERTED LINE
/
```

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

```
-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 file MYFILE.LIS contains the following lines:

```
                        1 ONE
                        2 TWO
;21-MAR                 .1 INSERTED LINE
```

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

## 2 How SUMSLP Processes Files

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

1. The editing commands are merged into a single stream in ascending order according to the value of locator1 (as described in Section 3.2). All edits that do not overlap or conflict with any other edits are applied to the source file without any further processing.

2. Editing commands that do conflict are resolved according to the precedence of the SUMSLP command file in which the commands occur. Precedence of SUMSLP command files is determined by the position of the file specifications following /UPDATE. The file specification listed last after /UPDATE has the highest precedence.

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

   An operation that deletes or replaces a line will affect not only the specified line, but also any lower precedence inserts or replacements to the same line. A deletion that specifies a range of lines (for example, -10,15) will delete all lines occurring in that range, including inserted lines from SUMSLP command files of lower precedence.

## 3 Specifying SUMSLP Editing Commands

SUMSLP editing commands let you update source files by adding, deleting, and replacing lines in a file. These commands contain certain characters that SUMSLP interprets as operators. This section first describes these operators and the general form for specifying SUMSLP editing commands. It then describes the editing commands used for specific editing functions.

## 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) | First character of a SUMSLP editing command |
| \ (backslash) | Suppresses audit trail generation |
| % (percent sign) | Reenables audit trail generation |
| / (slash) | Terminates the editing session |
| < (less-than character) | Escape character |

The percent sign (%) operator is used to reenable audit trail generation when generation has been suppressed by the backslash (\) operator.

The less-than character ( < ) operator is the escape character that lets you enter (in column 1) characters that SUMSLP would otherwise interpret as operators. For example, </ hides the slash character from SUMSLP, thereby enabling you to enter the slash into the output file without terminating the SUMSLP editing session. You can use the less-than character as an escape character for all SUMSLP operators listed in Table SUM–1 (including itself).

## 3.2 General Form of an Editing Command

The general form of a SUMSLP editing command is as follows:

```
-locator1 ,locator2\opt ,/audttrail/\opt ;comment\opt
inputline
    .
    .
    .
```

### Command Parameters

*– (minus sign)*

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

*locator1*

A line locator that causes SUMSLP 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 SUMSLP reads the next line in the editing command file. This field can be specified using any of the locator forms described below.

*locator2*

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/*

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 ( / ). If the editing command does not contain two locator fields, the audit trail specification must be preceded by two commas.

Audit trails generated by SUMSLP use the first character of the specified string as a delimiter. Usually, the first character of the audit trail is set to match the comment delimiter of the source file being edited.

An audit trail is produced automatically unless it has been suppressed through use of the backslash ( \ ) character. The default audit trail is ;**NEW** for each new line. The audit trail field will also contain a marker to indicate the number of lines deleted or replaced from the original file. The marker is placed on the first original line following a deletion and has the form –n, where n is the number of lines deleted.

*;comment*

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

*inputline*

A line of new text to be inserted into the file immediately following the current line. You can enter any number of input lines.

All fields in the command line are position-dependent; commas must be included as specified above.

The locator fields can take one of the following forms:

```
{       number      }
{         .         }
```

### Locator Field Parameters

*number*

A sequence number in the range of 1 through 9999 to which the current line pointer is moved.

*. (period)*

A period represents the current line.

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

SUMSLP 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.

The line numbers of inserted lines are distinguished from those of original lines by being preceded by a period. Inserted line numbers begin with .1 at the start of each group of new lines.

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

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

```
ABC
DEF
GHI
KLM
123456789
456
789
CBA
XYX
987
```

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

```
-5,,/;**07-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
```

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

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

SUMSLP has applied sequence numbers to the lines and added an audit trail to the line following line 5.

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

```
-4,,/;**08-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 files MYFILE.TST;3 and NEWFILE.LIS;2. NEWFILE.LIS;2 contains the following lines:

```
                    1 ABC
                    2 DEF
                    3 GHI
                    4 KLM
;**08-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
```

Again, SUMSLP has numbered the lines in sequence; this time the new input line is inserted after line 4.

## 3.4 Deleting Lines from a File

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

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

The locator1 and locator2 fields can take any of the forms described in Section 3.2. 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 example below shows how to delete lines from a file using SUMSLP. The input source file (TESTFILE.TST) consists of the following lines:

```
ABC
DEF
GHI
KLM
123456789
456
789
CBA
XYX
987
```

The SUMSLP update file (CHANGE01.UPD) for this example contains the following entries:

```
-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 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 for the above example, this next example shows how to delete a single line using the period locator. The new update file (CHANGE02.UPD) contains the following entries:

```
-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.

## 3.5 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 two-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, which, for replacement operations, is the line where new text is to be inserted.

For example, the command –4,8 instructs SUMSLP to move the line pointer to line 4 and replace line 4 and the next four lines (as represented by 8) with new input lines that immediately follow the command line.

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 lines:

```
ABC
DEF
GHI
123456789
BCN
CRB
BUR
```

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

```
-2,3,/;**LINE*REPLACED**/
XXXXXX
YYYYYY
/
```

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
-2                      4 123456789
                        5 BCN
                        6 CRB
                        7 BUR
```

## 3.6 Specifying the Audit Trail Text

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

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

Here newtrail is the new value (text) of the audit trail. All subsequent lines added will include the new audit trail text. All lines that indicate where lines have been deleted will include the first character of the new audit trail text as their first character. For example, if you specify the new audit trail JANUARY, the audit trail indicating a replaced line will be J**-2.

When you create a new audit trail, you may want to set the first character of the string to correspond to the comment delimiter that is used in the source file.

## 4 Running SUMSLP From a Command File

To run SUMSLP from a command file, generate a file that contains the required SUMSLP editing commands, one per line (see Section 3), input lines, and the SUMSLP terminator ( / ) in the first column of the last line.

UPDATE.COM contents:

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

You can execute the command file by using the Execute Procedure (@) command, as follows:

```
$ @UPDATE
```

Because the file type is the default file type COM, it can be omitted on the DCL command line. (See the *Guide to Using DCL and Command Procedures on VAX/VMS* 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 *VAX/VMS System Messages and Recovery Procedures Reference Manual* lists the diagnostic messages issued by SUMSLP and provides explanations and suggested user actions for these messages.

## COMMAND QUALIFIERS

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

**/LISTING***[=file-spec]*

## qualifier value

***file-spec***
The file specification of the listing file.

## DESCRIPTION

By default, there is no listing file. If you do not include a file specification, the listing file takes the same name as the input file, with a file type of LIS. You can specify another file type for the listing file, but LIS is the default. The listing file is described in Section 1.4.

/LISTING is a command qualifier only.

# /OUTPUT

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

## FORMAT

/OUTPUT*[=file-spec]*

### qualifier value

***file-spec***
The file specification of the output file.

## DESCRIPTION

If you do not specify a file, the output file has the same name and type as the input file, with a version number one higher than the highest existing version. The output file is described in Section 1.3.

/OUTPUT is a command qualifier only.

# /HEADER

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

## FORMAT

**/HEADER/OUTPUT**=*file-spec*

**qualifier value**

***/OUTPUT=file-spec***
The file specification of the output file.

## POSITIONAL QUALIFIER

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

# /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[,...])]*

### qualifier value

***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, they must be separated by commas, and the list must be enclosed in parentheses. The default file type of these files is initially 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 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 will not attempt to find an update file, but will generate any specified output or listing file. Enter the EDIT/SUM command with the /LISTING qualifier but without the /UPDATE qualifier to generate a numbered listing of your source program.

/UPDATE is a file qualifier only.

## EXAMPLES

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

The input source file FILE1.MAR is updated with the SUMSLP update file FILE1.UPD.

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

The input source file FILE2.MAR is updated with the SUMSLP update file UPD2.UPD.

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

The input source file FILE3.MAR is updated with the merged contents of SUMSLP update files UPD3A.UPD, UPD3B.ENH, and UPD3C.ENH. The editing commands in the three command files are applied according to the rules given in Section 2.

# Index

# Q



# T



# U

# READER'S COMMENTS

**Note:** This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

_____

_____

_____

Did you find errors in this manual? If so, specify the error and the page number.

_____

_____

_____

_____

_____

_____

_____

_____

_____

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
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

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

SSG PUBLICATIONS ZK1–3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062–2698

Do Not Tear - Fold Here

Cut Along Dotted Line