PDP–11 Keypad Editor User's Guide

Order Number AA–H853C–TC

August 1991

This document tells you how to use your terminal and the PDP–11 Keypad Editor (KED) to write, edit, and inspect any kind of text file and to store that file in one of the volumes on your system.

Whether you are a first-time or a long-time user of KED, you should begin with Chapter 1. This document has been completely rewritten to be more tutorial and task oriented.

Revision/Update Information:	This document supersedes the *PDP–11 Keypad Editor User's Guide*, AA–H853B–TC.
Operating System and Version:	RT–11 Version 5.6

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, March 1980
Updated, February 1981
Updated, July 1984
Revised, August 1989
Reprinted, August 1991

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.

Any 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 or equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013.

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

The following are trademarks of Digital Equipment Corporation:

CTS–300
DDCMP
DEC
DECmate
DECnet
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL
MASSBUS
MicroPDP–11
Micro/RSX
MicroVMS
PDP
P/OS
Professional
Q-bus
Rainbow
RSTS
RSX
RT–11
RTEM–11
UNIBUS
VAX
VAXcluster
VAX DOCUMENT
VMS
VT
Work Processor

digital™

S1763

This document was prepared using VAX DOCUMENT, Version 1.2.

Chapter 1
Introduction

An editor program lets you use your terminal to:

- Write, edit, and inspect any kind of text file (for example, a computer program, chapter in a book, table or figure, or list of names and addresses).
- Store that file in one of the volumes on your system.

This document describes the PDP–11 Keypad Editor (KED), which can be used on the RT–11 operating system. Information in this document is summarized in the *PDP–11 Keypad Editor Reference Card*.

KED is an interactive program. You type in text, use keypad function keys, and enter commands. KED displays your text and performs the editing operations you specify. Then it waits for more text and further instructions.

In this document, the interactive relationship between you and KED is called a work session.

1.1 Beginning and Ending Work Sessions—Chapter 2

Chapter 2 shows you how to:

- Begin a work session to create a file, edit a file, or inspect a file.
- Set your terminal to best suit the work you want to do.
- End a work session.

1.2 Writing with KED—Chapter 3

Under KED, your keyboard responds in the manner of an electric typewriter. For example, if you:

- Press a character key (letter, number, punctuation mark, or symbol), KED displays the character at the point where the cursor is located.
- Press RETURN (the return key), KED moves the cursor to the beginning of a new line.
- Press ⌫ (the delete key), KED deletes the character preceding the cursor.
- Press the space bar or TAB (the tab key), KED moves the cursor the appropriate number of spaces to the right.

As you type, KED continues to display your text. When you end your work session, KED stores the accumulated text as a file on the volume you specify.

Chapter 3 describes the ways you can use your terminal to type in text and perform text entry operations.

1.3 Editing with KED—Chapters 4 and 5

In addition to giving you the operations of an electric typewriter, KED lets you use your keyboard and keypad to edit (change) whatever you have written in a text file.

Every KED editing operation involves two steps:

1. Moving a cursor to the point in your text where you want to make a change
2. Making the change

1.3.1 Moving the Cursor—Chapter 4

The cursor is a blinking symbol, which shows where you are located in the file.

Some KED function keys and commands let you move the cursor by character, word, line, section, or page. They also let you move the cursor to any character in your text.

The keys you type and the commands you enter to move the cursor are described in Chapter 4.

1.3.2 Making Changes—Chapter 5

After you move the cursor to the proper location in your text, you can use KED function keys and commands to:

- Delete and restore a character, word, line, or larger section of text.
- Remove a section from one place in your text and insert it in another.
- Replace a character, character string (two or more contiguous characters), or multiple occurrences of a character string with another character or string that you specify.
- Convert letter characters from one case to the other (uppercase to lowercase, lowercase to uppercase).

Those and other KED editing operations are described in Chapter 5.

1.4 Using Macros and Command Files—Chapter 6

Since editing operations can consist of a sequence of KED functions and commands repeated over and over, KED lets you create macros (sequences of functions and commands) that you can use as needed.

KED also lets you create command files (sequences of commands) that you can use to preset KED or to complete a work session without typing any keys.

Chapter 6 tells you how to create, store, and use KED macros and command files.

1.5 Using Auxiliary Files—Chapters 7 and 8

To save time while editing text information that is contained in more than one file, KED lets you move text from one file to another. Using KED, you can:

- Move all or part of the file you are editing into an auxiliary file.
- Move all or part of an auxiliary file into any part of the file you are editing.

Those operations are described in Chapters 7 and 8.

1.6 Using Journal Files to Protect Your Work Session—Chapter 9

If you begin your work session as described in Chapter 2 and your system crashes or another malfunction has the same effect, you will lose all the work you performed. But if you begin your work session as described in Chapter 9, you can use a journal file to recover all or most of the work you performed and then continue with your work session.

1.7 Special Operations for Programmers—Chapter 10

In addition to the general-purpose writing and editing operations described in Chapters 2 through 9, KED gives you some special operations for program development:

1. Two ways to insert the escape character
2. For MACRO–11 programs, a renumbering operation that reorders your local symbols
3. For any program you write in a structured language, a structured tab operation that can:

   Insert statements
   Create multiple levels of indentation
   Change the indentation level of a group of statements
   Rewrite indentations to use the most efficient combination of tabs and spaces

Those operations are described in Chapter 10.

1.8 Special Procedures for the RT–11 Operating System—Appendix A

Chapters 2 through 10 describe procedures that are specific to KED. Appendix A describes procedures that are specific to the RT–11 operating system.

Appendix A gives you:

- Installation information
- Instructions for using the R, RUN, FRUN, and SRUN commands and the CSI file specification string

- Specific RT–11 information
- Instructions for using multiple editors and for printing text without ending your work session

1.9 Reference Appendixes

Appendix B lists command line and file specification options.

Appendix C lists KED functions and commands.

Appendix D lists decimal equivalent values for nonprinting ASCII characters.

Appendix E contains a work session exercise.

Appendix F lists KED error and help messages with explanations and problem solutions.

1.10 Your Reading Path

If you have never used KED, you will get the most out of this document by reading Chapters 2 through 10 (in that order).

If you have acquired a significant amount of experience with KED, you may want to examine the table of contents or the index in this document, and then read only the sections in this document that apply to your immediate needs.

Regardless of your experience with KED, you may also want to examine Appendix A before reading Chapters 2 through 10. Appendix A contains information that relates KED to your operating system.

Chapter 2

Beginning and Ending Your Work Sessions

This chapter shows you how to:

- Begin a work session and specify the required files (Section 2.1).
- Use the KED function keys and issue KED commands to set up your terminal (Sections 2.2 and 2.3).
- Write and edit some text (Section 2.4).
- End a work session (Section 2.5).

If you make a mistake that KED can detect, KED:

> Displays an appropriate message on your terminal.
> or
> Sounds a beeper on your terminal. If you press PF2 (the help key) on your keypad, KED then displays an appropriate message.

If that message does not give you enough information to correct your mistake, you can look the message up in Appendix F. There you will find an explanation of the message and instructions for correcting your mistake.

2.1 Beginning a Work Session

A complete command line for beginning a work session has the general format:

EDIT[/option] filespec[/option]

filespec	should be a standard file specification for input and output files in the form *[dev:]filnam.typ*. An input file contains text that you want to edit or inspect. An output file will contain the results of your work—the text you typed in or edited.
/option	specifies the kind of work you want to do or the kind of work you want KED to do. Appendix B describes all the command line options you can use in a work session.

The exact format of the command line you enter is determined by the kind of work you want to do. The following subsections show you how to begin the work you can do with KED.

2.1.1 Creating a New Text File

A command line for creating a new text file has the general format:

EDIT/CREATE output-filespec

/CREATE can be abbreviated to /CREA.

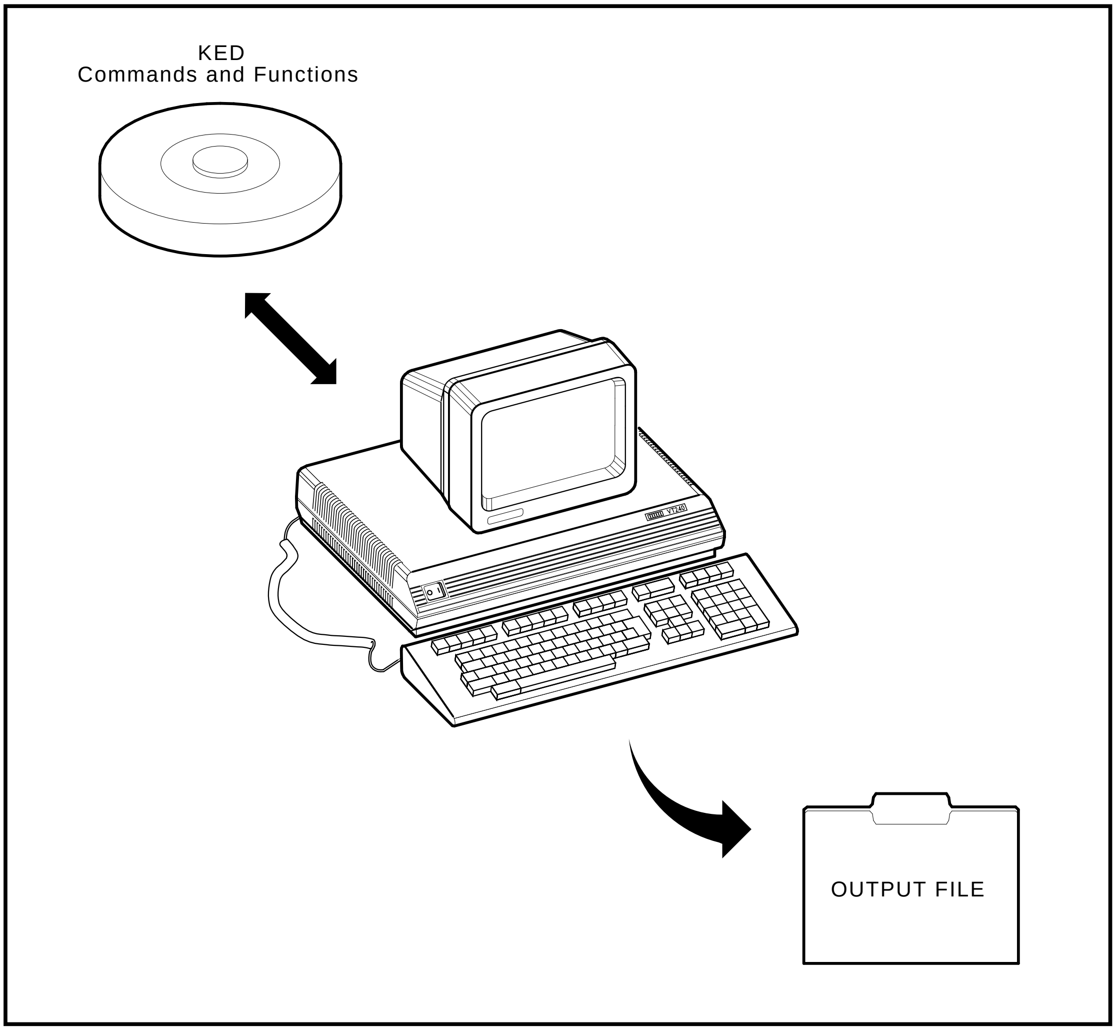
output-filespec is the specification for a new file.

If *output-filespec* exists, KED asks if you want to replace it. If that file exists and is protected, KED gives you the opportunity to inspect it.

For example, type:

```
EDIT/CREATE WEEK.TXT RET
```

KED creates output file WEEK.TXT, clears your terminal screen, and places its cursor in the upper left corner of your screen. Your terminal may use a block cursor. This document uses an underline cursor to show your location in a file.



MLO-003499

You are now ready to type in text as described in Section 2.4.1.

2.1.2 Editing an Existing Text File

A command line for editing an existing text file has the general format:

EDIT input-filespec

input-filespec — is the specification for the file that contains the text you want to edit. The default input file type is MAC.

If KED cannot find the file you specify, it asks for permission to create that file. If that file exists but is protected, KED gives you the opportunity to inspect it.

For example, to edit whatever you might have stored in file WEEK.TXT, type:

```
EDIT WEEK.TXT RET
```

KED displays your text and places its cursor in the upper left corner of the screen, which, in this case, is on the first character of the first word in WEEK.TXT.

In some cases, you may want to specify a new file to contain the output of your work session and keep the original text in the input file. A command line to do that has the general format:

EDIT/OUTPUT:output-filespec input-filespec

/OUTPUT — can be abbreviated to /OUTP.

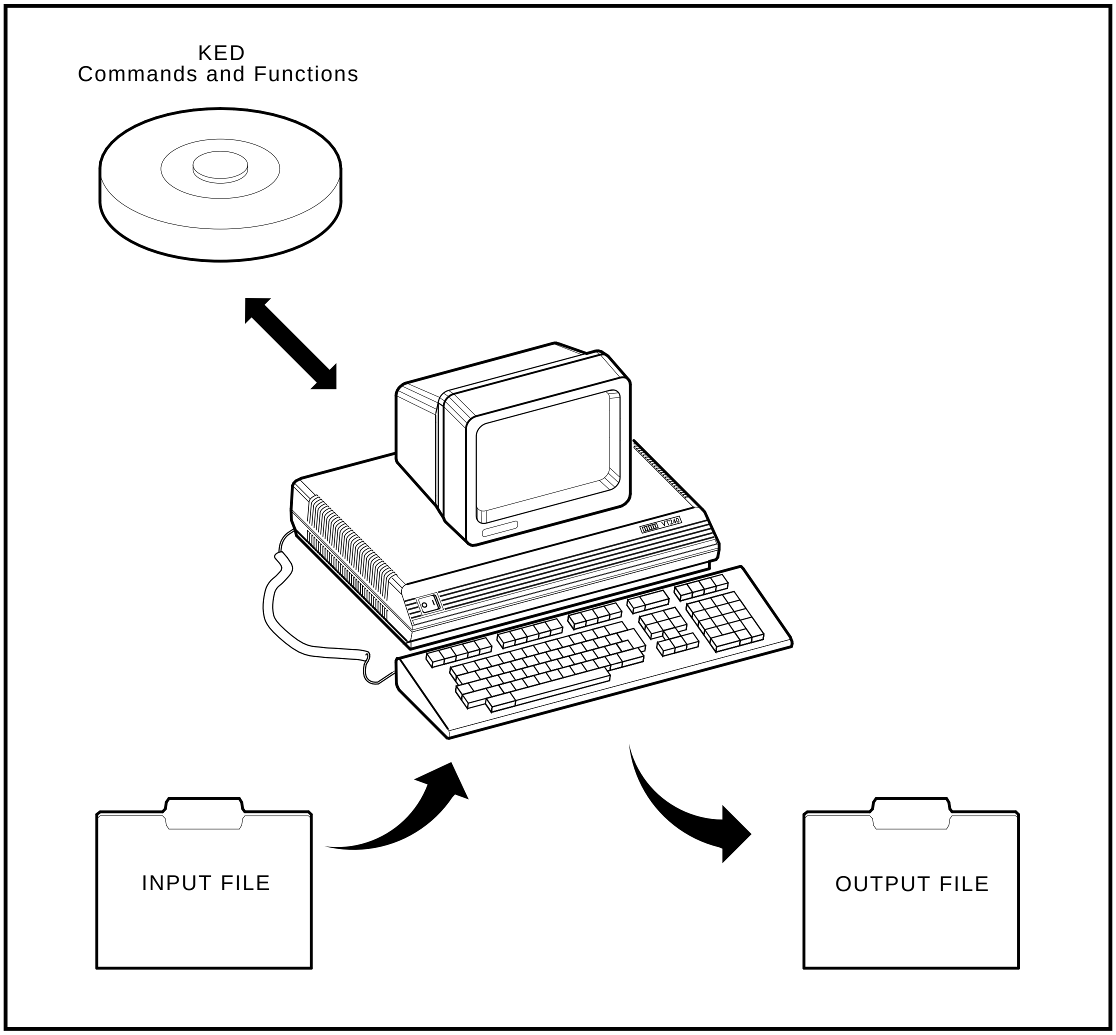
output-filespec — is the specification for the file that will contain the revised text.

input-filespec — is the specification for the file that contains the text you want to edit. At the end of your work session, that file is not altered.

For example, type:

```
EDIT/OUTPUT:WEEK1.TXT WEEK.TXT RET
```

KED displays WEEK.TXT. At the end of your work session, WEEK1.TXT will contain the revised text and WEEK.TXT is not altered.



MLO-003500

You are now ready to edit your text as described in Section 2.4.2.

2.1.3 Specifying a Maximum Output File Size

You should specify a maximum output file size when:

- You want to create an output file of a certain maximum size.
- You want to override the limit that KED places on the size of your output file in relation to the amount of free space available on your output volume, as described in Appendix A.

A command line for specifying a maximum output file size has the general format:

```
EDIT/CREATE  filespec/ALLOCATE:size
```

or

```
EDIT  filespec/ALLOCATE:size
```

/CREATE can be abbreviated to /CREA.

filespec is the specification for the output file that will contain the text you want to create or the input file that contains the text you want to edit. The default input file type is MAC.

/ALLOCATE can be abbreviated to /ALLO.

size is the maximum size (number of blocks) for the output file KED creates to contain the results of your work session. Since the largest input file size KED can accept is 32767, that number is the largest you can use.

For example, to create a new text file and specify 196 blocks as the maximum size for that file, type:

```
EDIT/CREATE WEEK.TXT/ALLOCATE:196 RET
```

To edit an existing text file, type:

```
EDIT WEEK.TXT/ALLOCATE:196 RET
```

KED begins your work session to edit the text stored in file WEEK.TXT. The `ALLOCATE:196` option specifies 196 blocks as the maximum size for the output file that will contain the edited text.

If you want to edit an existing text file and specify a new file to contain the output of your work session, while keeping the unaltered text in your input file, you can use the /OUTPUT option with the /ALLOCATE option in the following general format:

EDIT/OUTPUT:output-filespec/ALLOCATE:size input-filespec

For example, type:

```
EDIT/OUTPUT:WEEK1.TXT/ALLOCATE:196 WEEK.TXT RET
```

KED begins your work session to edit the text stored in file WEEK.TXT. The `ALLOCATE:196` option specifies 196 blocks as the maximum size for the new output file, WEEK1.TXT.

You cannot use the /OUTPUT option with the /CREATE option.

2.1.4 Inspecting an Existing Text File

A command line for inspecting an existing text file has the general format:

EDIT/INSPECT input-filespec

or

EDIT/READONLY input-filespec

/INSPECT can be abbreviated to /INSP.

/READONLY can be abbreviated to /READ.

input-filespec	is the specification for the file containing the text you want to inspect. The default file type is LST.

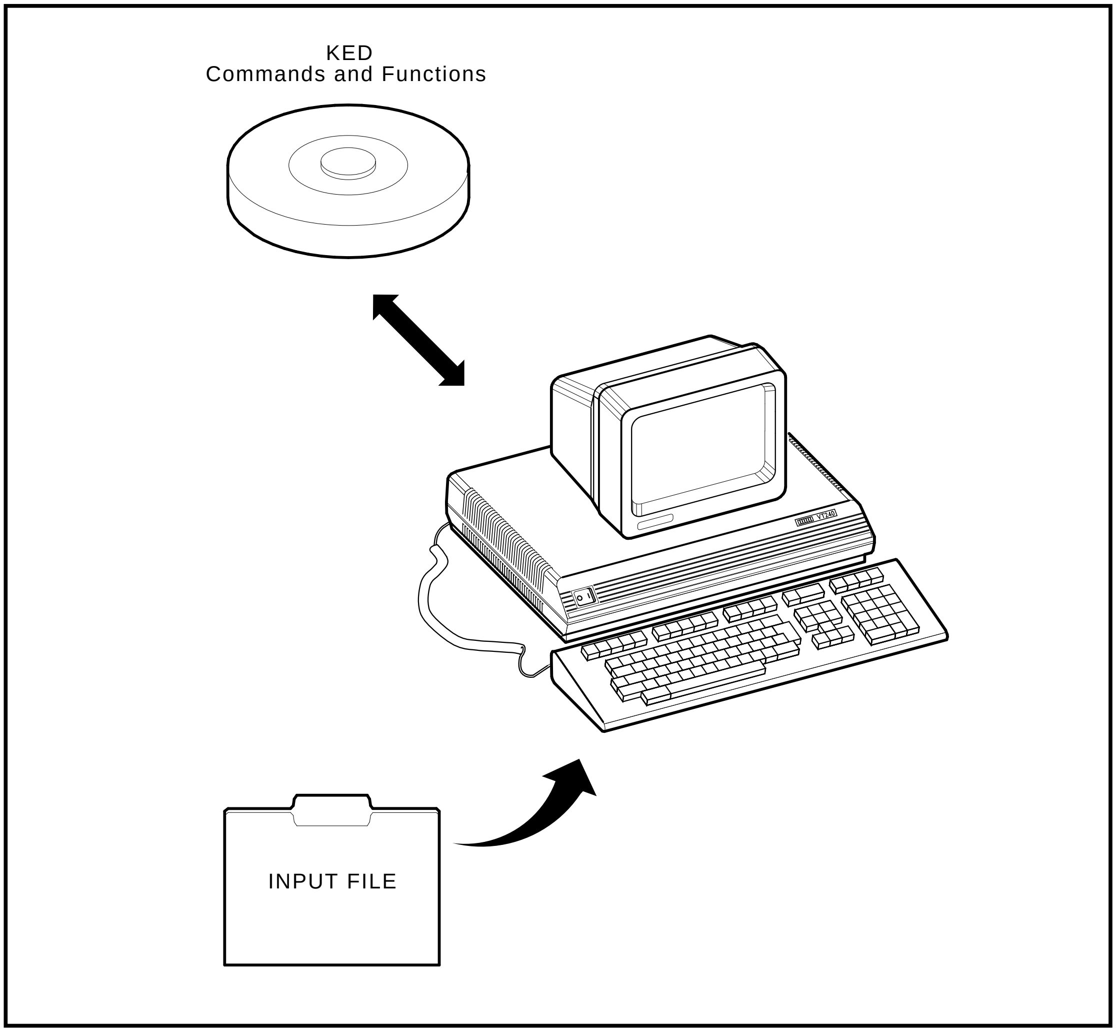
For example, type:

```
EDIT/INSPECT WEEK.TXT RET
```

or

```
EDIT/READONLY WEEK.TXT RET
```

KED disables its editing functions so you cannot accidentally change the text file you are inspecting. But you can use some of the KED function keys and commands to move around inside that file (see Chapter 4) and to create auxiliary output files (see Chapter 7).



MLO-003501

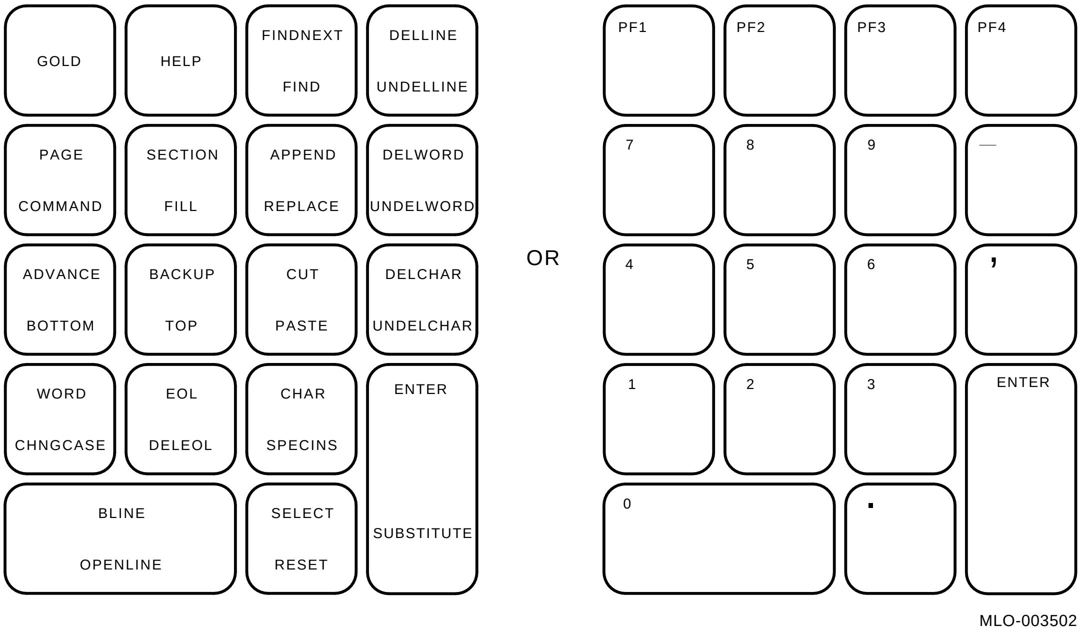
When you end your inspection as described in Section 2.5, the file WEEK.TXT is not altered.

2.2 Using the Function Keys and Issuing Commands

After beginning a work session to create or edit a text file (as described in Section 2.1.1, 2.1.2, or 2.1.3), you can tell KED what to do by pressing KED function keys on your keypad and keyboard and by issuing KED commands.

2.2.1 Keypad Function Keys

The KED function keys on your keypad are shown below.



Most of the function keys specify two functions: a standard function and an alternate function.

The standard function appears in the upper half of the key.
The alternate function appears in the lower half of the key.

You select the standard function by pressing the function key. The GOLD (PF1) key in the upper left corner of the keypad lets you select the alternate function on each function key. For example:

To delete a word of text, press DELWORD.
To insert that word back into the text, press GOLD and then press UNDELWORD.

You can cancel the effect of the GOLD key by pressing RESET.

2.2.2 The HELP Keypad Key

The HELP key displays a full-screen diagram of the keypad on your terminal. When that diagram is displayed:

1. If you press HELP again, you will see a full-screen list of KED commands, their abbreviations, and their operations.

2. If you continue to press HELP, you will see more full-screen lists describing the rest of the KED commands and all the GOLD keyboard functions. Eventually you will return to the keypad diagram.

You can return to your file at any time in that sequence by pressing ENTER on your keypad or CTRL/W on your keyboard.

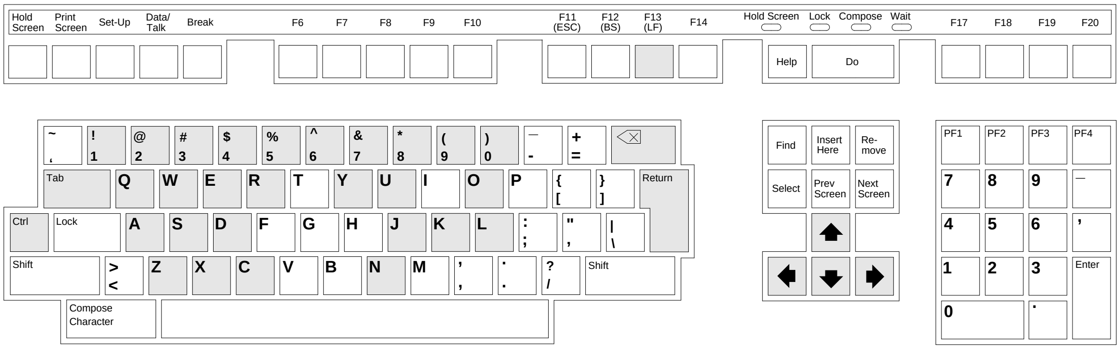
You can suppress all the full-screen HELP displays listed above by issuing the SET NOHELP command, as described in Section 2.3.4.

Even with SET NOHELP in effect, the HELP key can display an explanation of a KED error signal, as described in Appendix F.

You can use HELP anytime, except when KED is displaying one of its prompts.

2.2.3 Keyboard Function Keys

Other KED function keys on your keyboard are shown below.



MLO-003503

You will find out how to use those keys as you progress through this document.

Some keys are used together (for example, CTRL and C).
Some keys are used individually (for example, RETURN or TAB).
Some keys are used with the GOLD keypad key (for example, GOLD and a number).

2.2.4 Commands

To issue a KED command:

1. Press GOLD and COMMAND (first press GOLD, then press COMMAND).

   KED displays its command prompt.

   ```
   Command:
   ```

2. Type your command in response to that prompt.

   ```
   Command: your-command
   ```

You can use uppercase or lowercase letters in your command. Examples in this document use uppercase letters to indicate the parts you must type as shown and lowercase letters to indicate the parts you must supply (the variables).

3. Press ENTER or RETURN.

   KED responds to your command by performing the operation you specified.

   - If you press ENTER, KED erases your command and its command prompt.
   - If you press RETURN, KED erases your command but continues to display its command prompt so you can issue another command by repeating steps 2 and 3.

In other words:

GOLD COMMAND `your-command` ENTER
or
GOLD COMMAND `your-command` RET

If you obtain the command prompt and then decide you do not want to issue a command:

Press ENTER.
or
Type `CHANGE` and press ENTER.

KED will respond by erasing its command prompt.

If you obtain the command prompt, enter anything in response to the prompt, and then decide you do not want to issue a command, press CTRL/C. KED will respond by erasing its command prompt.

If you type a complete command and then decide you want to issue a different command, press CTRL/U. KED will respond by deleting the command you typed.

2.3 Commands and Functions You Can Use After Beginning Your Work Session

This section tells you how to:

Set your terminal screen to display longer text lines (Section 2.3.1).
Set the screen background to light or dark (Section 2.3.2).
Set the terminal error signal (Section 2.3.3).
Suppress the HELP display (Section 2.3.4).
Restore the screen display after an interruption (Section 2.3.5).
Stop an operation in progress (Section 2.3.6).

The commands described in this section can also be placed in a work-session initialization file as described in Sections 6.6 and 6.7, to prepare your work session environment (to give you the characteristics and operations you desire).

Appendix C describes all the functions and commands you can use in a work session. For a complete description of the function and commands you can use to move the cursor and change your text, see Chapter 4 and Chapter 5.

2.3.1 Setting Your Terminal to Display Longer Text Lines

By default (unless you change something), KED displays up to 78 characters in a single line. If the text you want to create, edit, or inspect contains lines that are more than 78 characters long, you can set KED to display up to 130 characters in a single line if you set your terminal for 132 characters.

To change the terminal display from 80 characters to 130 characters:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SCREEN] 132
   ```

   You do not have to type `SCREEN`.
3. Press ENTER.

   KED erases the screen and restores it with enough space for the display of 130 characters on a line.

In other words:

GOLD COMMAND `SET 132` ENTER

To return to the default (80 character) setting:

GOLD COMMAND `SET 80` ENTER

2.3.2 Setting the Terminal Screen Background to Light or Dark

You can set KED to display dark text on a light background or light text on a dark background.

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SCREEN] background
   ```

   You do not have to type `SCREEN`.

   `background` can be `LIGHT` or `DARK`.
3. Press ENTER.

   KED sets the text and background to dark/light or light/dark.

In other words:

GOLD COMMAND `SET LIGHT` ENTER
or
GOLD COMMAND `SET DARK` ENTER

2.3.3 Setting the Terminal Error Signal

By default, KED sounds the terminal beeper whenever a KED function or command fails. If you work in a quiet area, you can set KED to signal you silently by reversing the text and background.

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET QUIET
   ```

3. Press ENTER.

   The next time a KED function or command fails, KED signals you by reversing the text and background.

In other words:

```
GOLD COMMAND SET QUIET ENTER
```

You can see an explanation of the signal by pressing HELP. To continue your work session, press ENTER or any other valid function key.

To reset KED to sound the beeper:

```
GOLD COMMAND SET NOQUIET ENTER
```

2.3.4 Suppressing the HELP Displays

When you press HELP, by default, KED displays a full-screen diagram of the keypad, thus interrupting your work session. Once you become familiar with KED, you may not need that diagram and the other full-screen HELP displays, so KED gives you a way to suppress them.

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET NOHELP
   ```

3. Press ENTER.

   Then, when you press HELP, KED does not display a diagram of the keypad. Instead, it sounds the beeper on your terminal. If you again press HELP, KED tells you how to restore the full-screen HELP displays. In other words, the HELP function is disabled, except for the display of one-line error messages (KED's explanations of its error signals).

In other words:

```
GOLD COMMAND SET NOHELP ENTER
```

To reset KED to its default:

```
GOLD COMMAND SET HELP ENTER
```

2.3.5 Restoring the Screen Display After an Interruption

If the text that KED is displaying disappears or becomes corrupted in some way, press CTRL/R or CTRL/W.

If possible, KED restores your screen with a fresh image.

CTRL/R or CTRL/W can be useful when:

- You accidentally reset your terminal.
- KED displays a message or the keypad diagram after you press the HELP key.
- KED reverses the text and background to signal an error.
- Your operating system displays a message. In most cases, such messages cover part of the line that contains the KED cursor.
- Electronic interference generates extraneous characters on your screen while you are using a modem.
- The keypad keys display numbers and punctuation marks instead of performing KED functions.

You can use CTRL/R or CTRL/W anytime, except when KED is displaying one of its prompts. (If the keypad keys are displaying numbers and punctuation marks instead of performing KED functions, you can use CTRL/R or CTRL/W to restore the KED functions while KED is displaying a prompt, but the screen display will not be restored.)

2.3.6 Stopping an Operation in Progress

To stop an operation that KED is performing, press CTRL/C twice.

KED stops the operation, displays the part of your file that contains the cursor, and signals you by sounding a beeper or reversing the text and background on your terminal.

You can see an explanation of the signal by pressing HELP. To continue your work session, press ENTER or any other valid function key.

CTRL/C or two CTRL/Cs will not end your work session or destroy any files you are using.

2.4 Writing and Editing Text

2.4.1 Writing

After beginning a work session as described in Section 2.1.1, you continue the creation of WEEK.TXT by typing on the blank screen.

For example, type:

```
Monday was cold and rainy. RET
On Tuesday, it snowed. RET
Wednesday was sunny and warm. RET
```

KED displays the sentences as you type them, moving its cursor to the right of each character as it appears on your screen.

When you finish, you can end your work session and store WEEK.TXT as described in Section 2.5.

2.4.2 Editing

After beginning a work session as described in Section 2.1.2 or 2.1.3, you can edit WEEK.TXT.

For example, after typing the text described in Section 2.4.1, you decide to add another line of text at the end of your file.

1. Press GOLD and then press BOTTOM on your keypad.

   KED moves the cursor to an end-of-file marker that appears after the last line of text.

   ```
   Monday was cold and rainy.
   On Tuesday, it snowed.
   Wednesday was sunny and warm.

   _
   ```

2. Type:

   ```
   On Thursday, the lilacs bloomed.  RET
   ```

   KED displays the sentence as you type it. The cursor is now located at the end of your file.

   ```
   Monday was cold and rainy.
   On Tuesday, it snowed.
   Wednesday was sunny and warm.
   On Thursday, the lilacs bloomed.

   _
   ```

Next, you decide to change the word `warm` to `hot`.

1. Press BACKUP and then twice press EOL on your keypad.

   KED moves the cursor to the end of the previous line.

2. Press CTRL/J. On VT100 compatible terminals, you can press LF (the line-feed key) instead of CTRL/J.

   KED deletes the previous word (`warm`) and the period following it.

3. Type:

   ```
   hot.
   ```

   That gives you:

   ```
   Monday was cold and rainy.
   On Tuesday, it snowed.
   Wednesday was sunny and hot._
   On Thursday, the lilacs bloomed.
   ```

If you have made all the changes you wish to make, you are now ready to end your work session and store WEEK.TXT, as described in Section 2.5.

2.5 Ending a Work Session

You can end a work session while any part of a file is on your screen. You do not have to move the cursor to a special location.

At the end of a writing or editing session, you can save or discard your work.

2.5.1 Saving Your Work

To end your work session and save the text you created or edited:

1. Press GOLD and COMMAND on your keypad.
2. In response to the command prompt, type:

   ```
   EXIT
   ```

3. Press ENTER on your keypad.

   KED ends your work session, and your operating system displays its prompt.

In other words:

```
GOLD COMMAND EXIT ENTER
```

If you created a new text file, KED produces an output file that contains your work and stores that file in a default volume (disk or diskette) or in the volume you specified when you began your work session. That file has the same file name and file type as the output file you specified when you began your work session (for example, WEEK.TXT).

If you edited an existing text file, KED renames your input file, produces an output file, and stores both files in a default volume or in the volume you specified when you began your work session.

- The output file contains the revised version of your text. If you did not specify a different file in beginning your work session, that file has the same file name and file type as your input file (for example, WEEK.TXT).
- The renamed input file contains the unaltered version of your text. If you did not specify a different file when beginning your work session, that file has the same file name as your input file, but KED changes the file type to BAK to indicate it is a backup file (for example, WEEK.BAK).

If you inspected an existing text file, KED merely saves your input file (it does not produce any output files).

2.5.2 Discarding Your Work

To end your work session and discard whatever you created or edited:

1. Press GOLD and COMMAND on your keypad.

2. In response to the command prompt, type:

   ```
   QUIT
   ```

3. Press ENTER on your keypad.

   KED ends your work session, and your operating system displays its prompt.

In other words:

```
GOLD COMMAND QUIT ENTER
```

If you created a new text file, KED deletes that file and does not produce any output files.

If you edited an existing text file, KED saves the unaltered version of that file and does not produce any output files.

If you inspected an existing text file, KED does not discard anything (it merely saves that file).

2.6 What Next?

In addition to letting you use your terminal as an electric typewriter to create or edit text and store it in a file, KED lets you perform other text-entry operations that are described in Chapter 3. That chapter shows you how to:

Use your keyboard to insert text.
Insert nonprinting characters.
Use some special line terminators.
Set the right margin of your text.

Chapter 3
Writing with KED

This chapter describes the KED functions and commands you can use to write any kind of text file, such as a source program, a chapter in a book, a figure or table, or a list of names and addresses.

The keyboard (Section 3.1) contains the keys you use to insert text.

Certain function keys and commands let you perform additional text-entry operations:

- Inserting nonprinting characters (Section 3.2)
- Using special line terminators (Section 3.3)
- Setting the right margin (Section 3.4)

3.1 Using the Keyboard

The keyboard contains the keys you use to insert text:

Printing character keys
Shift key
Return key
Space bar
Tab key
Delete key

You cannot insert any text by pressing keys on the keypad.

Printing Character Keys

The printing character keys give you alphabetic characters, numeric characters, punctuation marks, and special symbols.

When you type a printing character key, KED inserts the character at the point where the cursor is located.

Shift Key

The shift key, SHIFT, lets you type alphabetic characters in uppercase and select the upper character on keys that give you two characters, as on a typewriter.

Return Key

The return key, RETURN, ends your current line of text and begins a new line.

Space Bar

The space bar lets you add spaces in a line of text.

Tab Key

The tab key, TAB, performs the same function as on a typewriter.

Delete Key

The delete key, ⌫, lets you delete any errors you make while typing in text or responding to KED prompts.

When you press ⌫, KED deletes the character preceding the cursor and stores it in the deleted character buffer.

3.2 Inserting Nonprinting Characters

Any text file you create with KED consists of two kinds of characters: printing characters and formatting characters.

- Printing characters include uppercase and lowercase alphabetic characters, numeric characters, punctuation marks, blank spaces, and special symbols, such as the dollar sign ($) and the asterisk (*).
- Formatting characters are nonprinting characters that cause your terminal to end the current line of text and begin a new line or to tabulate.

Appendix D contains a complete list of nonprinting characters and their decimal ASCII equivalents. The nonprinting characters you can use depends on your operating system.

To enter a decimal ASCII equivalent, use the SPECINS (special insertion) function.

1. Press GOLD.
2. Type the decimal ASCII equivalent, using the numeric keys on your keyboard.

   KED displays its `Repeat:` prompt and the number you typed.
3. Press GOLD and SPECINS.

   KED inserts the nonprinting character that corresponds to the decimal ASCII equivalent you typed.

In other words:

GOLD `number` GOLD SPECINS

Section 3.3.2 gives you an example of that procedure, which is used to insert a nonprinting line-feed character in a file.

3.3 Using Special Line Terminators

A text line consists of a string of printing characters followed by a nonprinting character or combination of nonprinting characters called a line terminator.

The most commonly used line terminator is the new-line terminator that KED inserts in your file each time you press RETURN. It consists of two nonprinting characters:

The carriage-return character, which moves the cursor to the left margin

The line-feed character, which moves the cursor down one line

Three additional line terminators are available for your use in formatting text:

The form-feed terminator (Section 3.3.1)
The line-feed terminator (Section 3.3.2)
The vertical-tab terminator (Section 3.3.3)

The form-feed terminator is often used, as it marks pagination in a document; it causes the printer to procede to the next page of text. The line-feed and vertical-tab terminators, however, are used only in rare circumstances.

3.3.1 Form-feed Terminator

A form-feed causes KED to start displaying characters at the left margin of the next line. A form-feed also causes most printers to cease printing text on the current sheet.

To insert a form-feed character in your text, press CTRL/L.

For example, type:

```
This is what CTRL/L a form feed does.
```

KED displays:

```
This is whatFF
a form feed does.
```

Typically, the text `This is what` would be printed as the last line on one sheet and the text `a form feed does` would be the first line printed on the next sheet.

3.3.2 Line-feed Terminator

A line-feed line terminator (not the LF keyboard key) causes KED to start displaying characters on the next line, directly below the place where you insert the line-feed character.

To insert a line-feed character in your text:

1. Press GOLD.
2. Type the decimal ASCII equivalent for the line-feed character (10).
3. Press GOLD and SPECINS.

In other words:

GOLD 10 GOLD SPECINS

For example:

1. Type:

   ```
   Half of this sentence
   ```

2. Type:

   GOLD 10 GOLD SPECINS

KED moves the cursor to the corresponding position in the next line down.

```
Half of this sentence
                     _
```

3. To complete the sentence, type:

```
                      is on the next line.
```

That gives you:

```
Half of this sentence
                      is on the next line._
```

3.3.3 Vertical-tab Terminator

A vertical-tab causes KED to start displaying characters at the left margin of the next line.

To insert a vertical-tab character in your text, press CTRL/K.

For example, type:

```
The vertical tab CTRL/K starts a new line.
```

KED displays:

```
The vertical tabVT
starts a new line.
```

3.4 Setting the Right Margin

You can set the right margin for new text (Section 3.4.1) or change the right margin for existing text (Section 3.4.2).

3.4.1 New Text

By default, KED lets you type any number of text lines, up to the limit that KED places on the size of your output file. When you reach the last character space on your screen, KED moves the cursor to the beginning of the next line, displays a continuation symbol (♦), and continues the line. To end the line before that point and begin a new one, you must press RETURN (or insert another line terminator described in Section 3.3).

If you wish, you can specify a maximum line length for the text you are creating. The operation is equivalent to setting an automatic right margin on a typewriter.

To specify a maximum line length:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET WRAP [number]
   ```

   `number` is the maximum number of characters that your new text lines will contain. You can specify any number from 1 to 256.

If you do not specify a number, KED uses a default of 78 if your screen width is 80 characters or a default of 130 if your screen width is 132 characters (that leaves 2 characters for a nonprinting line terminator).

3. Press ENTER.

In other words:

```
GOLD COMMAND SET WRAP [number] ENTER
```

As you type in text, KED inserts a nonprinting line terminator after the last complete word it can fit on the current line, and then begins a new line. Exceptions:

- If you type a single word that is longer than the maximum number of characters you specified for the right margin, KED leaves the word in a text line by itself.
- If you press TAB, KED processes it normally, and usually the text line that contains the horizontal-tab character reaches past the specified right margin.
- If you press RETURN, KED begins a new line.
- If you insert a carriage-return character by itself, KED overwrites the beginning of the text line that contains that character.

The SET WRAP command also gives you a convenient way to ensure that the text lines you type stay within the limits of your screen. For example, if your screen width is 80 characters, type:

```
GOLD COMMAND SET WRAP ENTER
```

KED sets a maximum line length of 78 characters (the default). When you reach the 78-character limit, KED inserts a line terminator after the last complete word that can fit on the line and then begins a new line.

To set a different maximum line length:

```
GOLD COMMAND SET WRAP new-number ENTER
```

To cancel the SET WRAP operation:

```
GOLD COMMAND SET NOWRAP ENTER
```

3.4.2 Existing Text

To change a right margin for existing text or set a right margin for unformatted text, use the SET WRAP command, the SELECT function, and the FILL function or command.

1. Issue the SET WRAP command to specify the line length.
   a. Press GOLD and COMMAND.
   b. In response to the command prompt, type:

      ```
      SET WRAP [number]
      ```

      `number` is the maximum number of characters the line will contain. You can specify any number from 1 to 256. (If you specify 1, word is placed on its own line.)

    If you do not specify a number, KED uses a default of 78 if your screen width is 80 characters or a default of 130 if your screen width is 132 characters (that leaves 2 characters for a nonprinting line terminator).

   c. Press ENTER.

2. Use the SELECT function to select the lines you want to reformat.

   a. Move the cursor (as described in Chapter 4) to the beginning of the first line you want to reformat.

   b. Press SELECT.

   c. Move the cursor to the end of the last line you want to reformat.

      As you move the cursor, KED highlights your selection by reversing the text and background or by underlining your selection. If you make a mistake in selecting lines, press GOLD and RESET.

3. Use the FILL function or command to reformat the lines you selected.

   To use the FILL function, press GOLD and FILL.

   To use the FILL command:

   a. Press GOLD and COMMAND.

   b. In response to the command prompt, type:

      `FILL`

   c. Press ENTER.

In other words:

GOLD COMMAND `SET WRAP [number]` ENTER
Move the cursor to the beginning of the first line.
SELECT
Move the cursor to the end of the last line.
Then:
GOLD FILL
or
GOLD COMMAND `FILL` ENTER

KED reformats the lines you selected according to the line length you specified. Specifically, KED:

- Inserts a single space in place of spaces and horizontal tabs.
- Substitutes a single space for carriage-return and line-feed characters, whether alone or combined as new-line terminators.

The fill operation joins paragraphs that are separated by blank lines created by carriage-return/line-feed combinations. To avoid having your paragraphs joined, process each paragraph with a separate FILL function or command or use vertical tabs (Section 3.3.3) or form feeds (Section 3.3.1) to separate your paragraphs.

The fill operation is not affected by the SET NOWRAP command. If you use a FILL function or command after using a SET NOWRAP command to disable a set wrap operation, KED continues to use the line length you specified with the SET WRAP command.

3.5 What Next?

Now that you have learned to use your terminal as a typewriter, to begin and end a work session, and to write any kind of text file, you are ready to explore the full power of KED to perform many useful editing operations.

First, however, you should learn all you need to know about moving the cursor, as described in Chapter 4. That chapter shows you how to move the cursor:

Character by character
Word by word
Line by line
Section by section
Page by page
To the end or beginning of your text file
Directly to any point in your text file

Chapter 4
Moving the Cursor

Every editing operation consists of two steps:

1. Moving the cursor to the point in your text file where you want to perform an operation
2. Performing that operation

This chapter describes the KED function keys and commands you can use to move the cursor. Chapter 5 describes the keys and commands you can use to perform editing operations.

KED lets you move the cursor:

- Character by character (Section 4.1)
- Word by word (Section 4.2)
- Line by line (Section 4.3)
- Section by section or page by page (Section 4.4)
- To the end or beginning of your text file (Section 4.5)
- Directly to any point in your text file (Section 4.6)

4.1 Moving by Character

Keyboard keys or keypad keys move the cursor one character at a time.

4.1.1 Left and Right Arrow Keyboard Keys

The $\boxed{\leftarrow}$ and $\boxed{\rightarrow}$ keys let you move the cursor from character to character.

Backing Up
To back up the cursor to the preceding character,
press $\boxed{\leftarrow}$.

Before:

```
123 45 6789
```

After:

```
123 45 6789
```

Advancing
To advance the cursor to the succeeding character, press $\boxed{\rightarrow}$.

Before:

```
123 45 6789
```

After:

```
123 45_6789
```

4.1.2 CHAR and BACKUP or ADVANCE Keypad Keys

The CHAR key moves the cursor one character at a time.

Backing Up

To back up the cursor to the preceding character, press BACKUP and CHAR.

Before:

```
Pawtucket, Rhode Island
```

After:

```
Pawtucket, Rhode Island
```

If you press CHAR again and again, KED continues to back up the cursor until you reverse direction by pressing ADVANCE.

Advancing

To advance the cursor to the succeeding character, press ADVANCE and CHAR.

Before:

```
Pawtucket, Rhode Island
```

After:

```
Pawtucket, Rhode Island
```

If you press CHAR again and again, KED continues to advance the cursor until you reverse direction by pressing BACKUP.

The ADVANCE and BACKUP keys affect keypad functions only. They do not affect the arrow keys.

4.2 Moving by Word

A word is a string of one or more characters (a character string) followed by one or more spaces. KED also considers a horizontal tab and all line terminators to be words.

The WORD and BACKUP or ADVANCE keypad keys move the cursor one word at a time.

Backing Up

To back up the cursor to the first character in the current word, press BACKUP and WORD.

Before:

```
East Mayhem, New Hampshire
```

After:

```
East Mayhem, New Hampshire
```

If you press WORD again and again, KED continues to back up the cursor until you reverse direction by pressing ADVANCE.

Advancing

To advance the cursor to the first character in the succeeding word, press ADVANCE and WORD.

Before:

```
East Mayhem, New Hampshire
```

After:

```
East Mayhem, New Hampshire
```

If you press WORD again and again, KED continues to advance the cursor until you reverse direction by pressing BACKUP.

4.3 Moving by Line

A line is one or more characters or words followed by a line terminator.

Keyboard keys and keypad keys move the cursor one line at a time.

4.3.1 Up and Down Arrow Keyboard Keys

The ↑ and ↓ keys let you move the cursor from line to line.

Backing Up

To back up the cursor to the corresponding character in the preceding line, press ↑ .

Before:

```
123 45
123 45 6789
```

After:

```
123 45
123 45 6789
```

If there is no corresponding character in the preceding line, KED puts the cursor at the end of that line.

Before:

```
123 45
123 45 6789
```

After:

```
123 45_
123 45 6789
```

Advancing

To advance the cursor to the corresponding character in the succeeding line, press ↓.

Before:

```
123 45 6789
123 45 6789
```

After:

```
123 45 6789
123 45 6789
```

If there is no corresponding character in the succeeding line, KED puts the cursor at the end of that line.

Before:

```
123 45 6789
123 45
```

After:

```
123 45 6789
123 45_
```

4.3.2 GOLD Keypad Key and Left-arrow or Right-arrow Keyboard Key

The GOLD keypad key and left-arrow or right-arrow keyboard key move the cursor to the beginning or end of a line. They are not affected by the ADVANCE and BACKUP keypad keys.

Backing Up

To back up the cursor to the beginning of the current line, press GOLD and ←.

Before:

```
33 Varnish Street
```

After:

```
33 Varnish Street
```

If the cursor is at the beginning of the line, KED backs it up to the beginning of the preceding line.

Before:

```
33 Varnish Street
Rumley, Maine
```

After:

```
33 Varnish Street
Rumley, Maine
```

If you press GOLD and ← again and again, KED continues to back up the cursor.

Advancing
To advance the cursor to the end of the current line, press GOLD and →.

Before:

```
33 Varnish Street
```

After:

```
33 Varnish Street_
```

If the cursor is at the end of the line, KED advances it to the end of the succeeding line.

Before:

```
33 Varnish Street_
Rumley, Maine
```

After:

```
33 Varnish Street
Rumley, Maine_
```

If you press GOLD and → again and again, KED continues to advance the cursor.

4.3.3 BLINE and EOL Keypad Keys

The BLINE key and the EOL key move the cursor line by line.

4.3.3.1 BLINE (Beginning of Line) Key

The BLINE key moves the cursor to the beginning of a line.

Backing Up
To back up the cursor to the beginning of the current line, press BACKUP and BLINE.

Before:

```
33 Varnish Street
```

After:

```
33 Varnish Street
```

If the cursor is at the beginning of the line, KED backs it up to the beginning of the preceding line.

Before:

```
33 Varnish Street
Rumley, Maine
```

After:

```
33 Varnish Street
Rumley, Maine
```

If you press BLINE again and again, KED continues to back up the cursor until you reverse direction by pressing ADVANCE.

Advancing

To advance the cursor to the beginning of the succeeding line, press ADVANCE and BLINE.

Before:

```
33 Varnish Street
Rumley, Maine
```

After:

```
33 Varnish Street
Rumley, Maine
```

If you press BLINE again and again, KED continues to advance the cursor until you reverse direction by pressing BACKUP.

The ADVANCE and BACKUP keys affect keypad functions only. They do not affect the arrow keys.

4.3.3.2 EOL (End of Line) Key

The EOL key moves the cursor to the end of a line.

Backing Up

To back up the cursor to the end of the preceding line, press BACKUP and EOL.

Before:

```
33 Varnish Street
Rumley, Maine
```

After:

```
33 Varnish Street_
Rumley, Maine
```

If you press EOL again and again, KED continues to back up the cursor until you reverse direction by pressing ADVANCE.

Advancing

To advance the cursor to the end of the current line, press ADVANCE and EOL.

Before:

```
33 Varnish Street
```

After:

```
33 Varnish Street_
```

If the cursor is at the end of the line, KED advances it to the end of the succeeding line.

Before:

```
33 Varnish Street_
Rumley, Maine
```

After:

```
33 Varnish Street
Rumley, Maine_
```

If you press EOL again and again, KED continues to advance the cursor until you reverse direction by pressing BACKUP.

The ADVANCE and BACKUP keys affect keypad functions only. They do not affect the arrow keys.

4.4 Moving by Section or Page

You can step through a text file in multiple-line jumps by:

Using the KED defaults (Section 4.4.1)
or
Specifying the length of each jump (Section 4.4.2)

4.4.1 Using the Defaults

The keypad SECTION and PAGE keys let you use the KED defaults in the following ways:

SECTION Key

By default, a section is 16 lines of text.

- To advance the cursor 16 lines, press ADVANCE and SECTION.

  KED moves the cursor forward to the beginning of the 17th line.

- To back up the cursor 16 lines, press BACKUP and SECTION.

  KED moves the cursor backward to the beginning of the 17th line.

PAGE Key

By default, a page is all the text between the cursor and the next form-feed character.

- To advance the cursor to the beginning of the succeeding page, press ADVANCE and PAGE.

  KED moves the cursor to the character at the beginning of the succeeding page.

- To back up the cursor to the beginning of the current page, press BACKUP and PAGE.

  If the cursor is at the beginning of a page, KED moves it to the beginning of the preceding page.

4.4.2 Specifying the Length of Sections and Pages

You can specify the length of a section and a page and then use the keypad SECTION and PAGE keys interchangeably.

For example, you can:

- Specify a section or page to be a certain number of lines of text. Each time you press SECTION or PAGE, KED moves the cursor that number of lines.
- Specify a section or page to be all the text between occurrences of a character or character string (a marker) that you place in your text file. Each time you press SECTION or PAGE, KED moves the cursor to the first occurrence of that marker.

4.4.2.1 By Number of Lines

To specify a section or page to be a certain number of lines of text and to move the cursor:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [ENTITY] SECTION number [LINES]
   ```

   or

   ```
   SET [ENTITY] PAGE number [LINES]
   ```

   `number` can be from 1 to 32767.
3. Press ENTER.
4. Press BACKUP or ADVANCE and SECTION or PAGE.

   KED moves the cursor the number of lines you specified.

Examples:

- To specify a section to be 35 lines of text:

  GOLD COMMAND `SET SECTION 35` ENTER

  Then, press BACKUP or ADVANCE and SECTION.
- To specify a page to be 100 lines of text:

  GOLD COMMAND `SET PAGE 100` ENTER

  Then, press BACKUP or ADVANCE and PAGE.

4.4.2.2 By Marker

To specify a section or page to be all the text between occurrences of a marker and to move the cursor:

1. If you do not want to use existing text as a marker, place an appropriate marker in your text file as many times as it will be needed.

Your marker can be a printing character, a string of printing characters, or any nonprinting line terminator. For a section, it can be no longer than 51 characters. For a page, it can be no longer than 53 characters.

By default, KED does not distinguish between uppercase and lowercase letters. To make KED recognize the case of letters, see Section 4.6.1.

2. Press GOLD and COMMAND.
3. In response to the command prompt, type:

   ```
   SET [ENTITY] SECTION marker
   ```

   or

   ```
   SET [ENTITY] PAGE marker
   ```

   `marker` is existing text or the character, character string, or line terminator you selected in step 1, enclosed by single (′) or double (") quotation marks.
4. Press ENTER.
5. Press BACKUP or ADVANCE and SECTION or PAGE.

   KED moves the cursor to the marker you specified.

Examples:

- To specify a section to be all the text between occurrences of the marker NEXTSECT:

  GOLD COMMAND `SET SECTION "NEXTSECT"` ENTER
  or
  GOLD COMMAND `SET SECTION ′NEXTSECT′` ENTER

  Then, press BACKUP or ADVANCE and SECTION.
- To specify a page to be all the text between occurrences of the marker NEXTPAGE:

  GOLD COMMAND `SET PAGE "NEXTPAGE"` ENTER
  or
  GOLD COMMAND `SET PAGE ′NEXTPAGE′` ENTER

  Then, press BACKUP or ADVANCE and PAGE.

4.5 Moving to the End or Beginning of Your Text File

Keypad and keyboard keys let you move the cursor to the end or beginning of your text file.

- To advance the cursor from its current location to the end of your file, press:

  GOLD and BOTTOM
  or
  GOLD and ↓

KED moves the cursor to the end-of-file marker at the end of your file.

- To back up the cursor from its current location to the beginning of your file, press:

  GOLD and TOP
  or
  GOLD and ↑

  KED moves the cursor to the beginning of your file.

When the cursor is far enough from the end or beginning of your file, KED temporarily displays the message `WORKING...` to let you know that nothing is wrong and that the cursor is still moving.

4.6 Moving Directly to Any Point in Your Text File

KED gives you a search operation that can move the cursor directly to any point in your text file.

To start your search:

1. Press GOLD and FIND.

   KED displays its model prompt.

   `Model:`

2. Type the character string you are looking for. It can be up to 60 characters long and can contain any printing character or any line terminator.

   You can press ⌫ (the delete key) or CTRL/U to correct any typing errors. If you press CTRL/C, KED deletes its model prompt and your character string and restores the screen display.

3. Press one of the following keys:

   ADVANCE or BACKUP to enter your search model, specify your search direction, and start the search.
   ENTER to enter your search model without starting the search. You must press FINDNEXT to start the search.
   ↑ to enter your search model and search backward without changing the keypad directional mode.
   ↓ to enter your search model and search forward without changing the keypad directional mode.

   KED places your character string in a search buffer, and then searches forward or backward from the current location of the cursor to the end or the beginning of your file. When the cursor is far enough from your character string, KED temporarily displays the message `WORKING...` to let you know that nothing is wrong and that the cursor is still moving.

In other words:

GOLD FIND `character-string` ADVANCE
or

GOLD FIND `character-string` BACKUP

or

GOLD FIND `character-string` ENTER

or

GOLD FIND `character-string` FINDNEXT

or

GOLD FIND `character-string` ↑

or

GOLD FIND `character-string` ↓

To continue your search or to start a search after pressing ENTER, press FINDNEXT.

- If you are at the end of your file or page, press BACKUP and FINDNEXT.
- If you are at the beginning of your file or page, press ADVANCE and FINDNEXT.

If KED finds your character string, it moves the cursor to the first character in that string. If KED does not find your character string, it signals you by sounding a beeper or reversing the text and background on your terminal. If you press HELP, KED displays:

```
Target not found
```

If you press GOLD and FIND and then decide you do not want to look for a character string, press ENTER. KED will respond by deleting its model prompt.

4.6.1 Distinguishing Between Uppercase and Lowercase Letters

By default, KED does not distinguish between uppercase and lowercase letters. For example, `SYMBOL` is the same as `Symbol` or `symbol`.

To set KED so it will distinguish between uppercase and lowercase letters:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SEARCH] EXACT
   ```

3. Press ENTER.

   KED will now distinguish between uppercase and lowercase letters. For example, `SYMBOL`, `Symbol`, `symbol` are now three different character strings.

In other words:

GOLD COMMAND `SET EXACT` ENTER

Then, start your search as described in Section 4.6.

To reset KED to its default:

GOLD COMMAND `SET GENERAL` ENTER

KED will no longer distinguish between uppercase and lowercase letters.

4.6.2 Moving to the End of Your Search String

By default, KED moves the cursor to the first character of the string it finds.

To set KED so it will move the cursor to the right of the last character in your search string:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SEARCH] END
   ```

3. Press ENTER.

   KED will now move the cursor to the right of the last character of the string it finds.

In other words:

GOLD COMMAND `SET END` ENTER

Then, start your search as described in Section 4.6.

To reset KED to its default:

GOLD COMMAND `SET BEGIN` ENTER

KED will now move the cursor to the first character of the string it finds.

4.6.3 Limiting the Area of Your Search

By default, KED continues a search to the end or beginning of your file.

To limit your search to the current page of text:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SEARCH] BOUNDED
   ```

3. Press ENTER.

   KED will now limit its search to the current page. By default, that page is all the text between the cursor and the next form-feed character. You may have used the SET ENTITY PAGE command to specify that page as a certain number of lines of text (Section 4.4.2.1) or as all the text between the cursor and a marker (Section 4.4.2.2).

In other words:

GOLD COMMAND `SET BOUNDED` ENTER

Then, start your search as described in Section 4.6.

- If you start your search by pressing ADVANCE, KED searches to the end of the page.
- If you start your search by pressing BACKUP, KED searches to the beginning of the page.

- If KED does not find your search string on the current page, it signals you by sounding a beeper or reversing the text and background on your terminal. If you press HELP, KED displays:

  ```
  Bounded search reached bound
  ```

To reset KED to its default:

```
GOLD COMMAND SET UNBOUNDED ENTER
```

4.6.4 Controlling Cursor Movement When Search String Is Not Found

By default, the cursor remains in its current position when your search string is not found.

To set KED so it will move the cursor to the end or beginning of your file when your search string is not found:

1. Press GOLD and COMMAND.
2. In response to the command prompt, type:

   ```
   SET [SEARCH] NORETURN
   ```

3. Press ENTER.

   KED will now move the cursor to the end or beginning of your file when your search string is not found.

In other words:

```
GOLD COMMAND SET NORETURN ENTER
```

Then, start your search as described in Section 4.6.

To reset KED to its default:

```
GOLD COMMAND SET RETURN ENTER
```

4.7 What Next?

Now that you know how to move the cursor to any point in your text file, you can make all the many useful changes described in Chapter 5. That chapter shows you how to:

Delete and restore a character.
Delete and restore a word.
Delete and restore a line of text.
Insert a blank line.
Delete a large section of text.
Insert a new section of text in multiple locations.
Remove a section of text from one place in your file and insert it in another place.
Empty the paste buffer.
Replace a section of text with another section.
Substitute a character string for multiple occurrences of another character string.

Change alphabetic characters from uppercase to lowercase or lowercase to uppercase.
Repeat KED editing operations.

Chapter 5
Making Changes

Every editing operation consists of two steps:

1. Moving the cursor to the point in your file where you want to perform an operation
2. Performing that operation

Chapter 4 describes the KED function keys and commands you can use to move the cursor. This chapter describes the keys and commands you can use to:

- Delete and restore a character, word, or line of text (Sections 5.1, 5.2, and 5.3)
- Insert a blank line (Section 5.4)
- Empty the paste buffer (Section 5.5)
- Delete and restore a large section of text (Section 5.6)
- Insert a new section of text in multiple locations (Section 5.7)
- Remove a section of text from one place in your file and insert it in another place (Section 5.8)
- Replace a section of text with another section (Section 5.9)
- Substitute a character string for multiple occurrences of another character string (Section 5.10)
- Force alphabetic characters to uppercase or lowercase (Section 5.11)
- Change alphabetic characters from uppercase to lowercase or lowercase to uppercase (Section 5.12)
- Repeat KED editing operations (Section 5.13)

To perform those operations, you must begin your work session to create or edit a file. You cannot make any changes while inspecting a file.

5.1 Deleting and Restoring a Character

A character is any character in the ASCII set listed in Appendix D, including the horizontal tab character, all line terminators, and all nonprinting characters.

To delete a character:

1. Move the cursor to the character you want to delete.
2. Press DELCHAR.

KED deletes the character, stores it in a character buffer, and closes up the resulting space.

Before:

```
emulatolr
```

After:

```
emulator
```

If you move the cursor to the line terminator, KED deletes that terminator and moves the next line up to the end of the line that contained the terminator.

Before:

```
emulator-_
system function
```

After:

```
emulator-system function
```

You can also use the ⌫ (delete key) to delete a character as described in Chapter 3.

The deleted character remains in the character buffer until the next time you press DELCHAR or ⌫, causing KED to store a new character in the buffer.

To restore that character to your text press GOLD and UNDELCHAR or GOLD and ⌫. KED inserts the character at the point where the cursor is located, shifting the remainder of the line one character to the right.

While responding to a KED prompt, you cannot restore a character you deleted unless you type it again.

5.2 Deleting and Restoring a Word

A word is one or more characters followed by one or more spaces. KED also considers the horizontal-tab character, all line terminators, and any spaces preceding a character to be words.

You can use the DELWORD keypad key or CTRL/J keyboard keys to delete a word. If you are using a VT100-compatible terminal, you can also use the line-feed (LF) keyboard key to delete a word.

Then, you can use the GOLD and UNDELWORD keypad keys or the GOLD keypad key and CTRL/J keyboard keys to restore that word to your text. If you are using a VT100-compatible terminal, you can also use the GOLD keypad key and line-feed (LF) keyboard key to restore that word to your text.

5.2.1 DELWORD Keypad Key

To delete a word or the part of a word following the cursor:

1. Move the cursor to the word or part of a word you want to delete.
2. Press DELWORD.

KED deletes the word or part of a word, stores it (up to 80 characters) in a word buffer, and closes up the resulting space.

Before:

```
red, blue, green, turnip, yellow
```

After:

```
red, blue, green, yellow
```

If you move the cursor inside a word, KED deletes only the part that follows the cursor.

Before:

```
red, blue, green, yellow
```

After:

```
red, blue, green, yell_
```

The deleted word remains in the word buffer until you press DELWORD, CTRL/J, or LF (the line-feed key), causing KED to store a new word in the buffer.

To restore that word to your text, press GOLD and UNDELWORD, GOLD and CTRL/J, or GOLD and LF.

KED inserts the word at the point where the cursor is located, shifting the remainder of the line the appropriate number of characters to the right.

5.2.2 CTRL/J Keyboard Keys or Line-feed (LF) Keyboard Key

To delete a word or the part of a word preceding the cursor:

1. Move the cursor to the right side of the word or part of a word you want to delete.
2. Press CTRL/J. If you are using a VT100-compatible terminal, you can press LF (the line-feed key) instead.

   KED deletes all the characters that precede the cursor up to the beginning of the word, stores those characters (up to 80) in a word buffer, and closes up the resulting space.

   Before:

   ```
   red, blue, green, turnip, yellow
   ```

   After:

   ```
   red, blue, green, yellow
   ```

   If you move the cursor inside a word, KED deletes only the part that precedes the cursor.

   Before:

   ```
   red, blue, green, yellow
   ```

   After:

```
red, blue, green, low
```

The deleted word remains in the word buffer until you press CTRL/J, LF (the line-feed key), or DELWORD, causing KED to store a new word in the buffer.

To restore that word to your text, press GOLD and UNDELWORD, GOLD and CTRL/J, or GOLD and LF.

KED inserts the word at the point where the cursor is located, shifting the remainder of the line the appropriate number of characters to the right.

5.3 Deleting and Restoring a Line

A line is one or more characters or words followed by a line terminator.

You can use the DELLINE or DELEOL keypad key or the CTRL/U keyboard keys to delete a line.

Then, you can use the GOLD and UNDELLINE keypad keys or the GOLD keypad key and CTRL/U keyboard keys to restore that line to your text.

5.3.1 DELLINE Keypad Key

To delete a line or the remainder of a line following the cursor:

1. Move the cursor to the first character or word in the line you want to delete.
2. Press DELLINE.

   KED deletes the line, stores it (up to 132 characters) in a line buffer, and closes up the resulting space.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   Rip Torn
   James Garfield
   ```

   After:

   ```
   Chester Arthur
   Millard Fillmore
   James Garfield
   ```

   If you move the cursor beyond the first character in a line, KED deletes all the characters between the cursor and the beginning of the next line, including the line terminator, and moves the next line up to the end of the line that contained the terminator.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   James Garfield
   ```

   After:

```
Chester Arthur
Millard James Garfield
```

The deleted line remains in the line buffer until you press DELLINE, CTRL/U, or GOLD and DELEOL, causing KED to store a new line in the buffer.

To restore that line to your text, press GOLD and UNDELLINE or GOLD and CTRL/U.

KED inserts the line at the point where the cursor is located, shifting the remaining lines down one line.

5.3.2 DELEOL Keypad Key

To delete a line or the remainder of a line following the cursor but not the line terminator:

1. Move the cursor to the first character or word in the line you want to delete.
2. Press GOLD and DELEOL.

   KED deletes the line, stores it (up to 132 characters) in a line buffer, but does not close up the resulting vertical space.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   Rip Torn
   James Garfield
   ```

   After:

   ```
   Chester Arthur
   Millard Fillmore

   James Garfield
   ```

   If you move the cursor beyond the first character in a line but not to the line terminator, KED deletes only the printing characters that remain in the line.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   Rip Torn
   James Garfield
   ```

   After:

   ```
   Chester Arthur
   Millard Fillmore
   Rip _
   James Garfield
   ```

   If you move the cursor to the line terminator, KED deletes the next line.

   Before:

```
Chester Arthur
Millard Fillmore
Rip Torn_
James Garfield
```

After:

```
Chester Arthur
Millard Fillmore
Rip Torn_
```

The deleted line remains in the line buffer until you press DELEOL, CTRL/U, or GOLD and DELLINE, causing KED to store a new line in the buffer.

To restore that line to your text, press GOLD and UNDELLINE or GOLD and CTRL/U.

KED inserts the line at the point where the cursor is located.

5.3.3 CTRL/U Keyboard Keys

To delete a line or the part of a line preceding the cursor:

1. Move the cursor to the right side of the line or part of a line you want to delete.
2. Press CTRL/U.

   KED deletes all the characters that precede the cursor up to the beginning of the line, stores those characters (up to 132) in a line buffer, but does not close up the resulting vertical space.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   Rip Torn_
   James Garfield
   ```

   After:

   ```
   Chester Arthur
   Millard Fillmore
   _
   James Garfield
   ```

   If you move the cursor to delete a part of the line, KED deletes only the part that precedes the cursor.

   Before:

   ```
   Chester Arthur
   Millard Fillmore
   Rip Torn
   James Garfield
   ```

   After:

   ```
   Chester Arthur
   Millard Fillmore
   Torn
   James Garfield
   ```

If you move the cursor to the beginning of a line, KED deletes the preceding line.

Before:

```
Chester Arthur
Millard Fillmore
Rip Torn
James Garfield
```

After:

```
Chester Arthur
Rip Torn
James Garfield
```

The deleted line remains in the line buffer until you press CTRL/U, DELLINE, or GOLD and DELEOL, causing KED to store a new line in the buffer.

To restore that line to your text, press GOLD and UNDELLINE or GOLD and CTRL/U.

KED inserts the line at the point where the cursor is located.

If you are responding to a KED prompt, CTRL/U deletes all the characters you typed, and you cannot restore them (except by typing them again).

5.4 Inserting a Blank Line

You can use the OPENLINE function to insert a blank line within a line of text or between two lines of text.

To insert a blank line within a line of text:

1. Move the cursor to the point in the text line where you want the blank line to be inserted.
2. Press RETURN and then press GOLD and OPENLINE.

   KED moves the rest of the text line to the next line down, inserts a blank line between the two text lines, and leaves the cursor at the beginning of the open text line.

   Before:

   ```
   Chester ArthurMillard Fillmore
   ```

   After:

   ```
   Chester Arthur

   Millard Fillmore
   ```

To delete the blank line, press DELLINE and ⌫.

To insert a blank line between two lines of text:

1. Move the cursor to the beginning of the second text line.
2. Press GOLD and OPENLINE.

KED inserts a blank line between the two text lines and moves the cursor to the beginning of the blank line.

Before:

```
Chester Arthur
Millard Fillmore
```

After:

```
Chester Arthur

Millard Fillmore
```

To delete the blank line, press DELLINE.

5.5 Emptying the Paste Buffer

You should empty the paste buffer before:

- Using the REPLACE function (Section 5.6) to delete a large section of text
- Using the SUBSTITUTE function (Section 5.10) to delete multiple occurrences of a character string

To empty the paste buffer:

1. Press GOLD and COMMAND.
2. Type:

   ```
   CLEAR PASTE
   ```

3. Press ENTER.

   KED deletes the contents of the paste buffer.

In other words:

GOLD COMMAND `CLEAR PASTE` ENTER

5.6 Deleting and Restoring Large Sections of Text

You can use the SELECT and CUT functions to delete a large section of text.

1. Select the section of text you want to delete.
   a. Move the cursor to the first character in that section.
   b. Press SELECT.
   c. Move the cursor to the character following the last character in that section.

      KED highlights your selection by underlining it or displaying it in reverse video (reversing the text and background).

If you make a mistake in selecting the section you want to delete, you can press GOLD and RESET or back up the cursor to the end of the section you want to delete.

2. Press CUT.

   KED deletes the text you selected (up to the limit of your system) and stores it in a paste buffer.

In other words:

Move the cursor to the first character.
SELECT
Move the cursor to the character following the last character.
CUT

To restore the text you deleted, press GOLD and PASTE.

KED inserts the contents of the paste buffer (the text you deleted) at the point where the cursor is located and moves the cursor to the first character following the inserted text.

The size of the paste buffer depends on your operating system, the monitor you are using, and the amount of memory in your system. The minimum size is 512 characters. If the paste buffer is too small to hold your selection, KED signals you, by sounding a beeper or reversing the text and background on your terminal, and does not perform the cut operation. If you press HELP, KED displays:

```
Select range too large for CUT
```

When the text you want to delete is too large for the paste buffer, you can use the CLEAR PASTE command and the SELECT and REPLACE functions instead of the SELECT and CUT functions. However, you cannot restore the deleted text unless you use a backup copy of your input file.

1. Empty the paste buffer as described in Section 5.5.
2. Use the SELECT function to select the text you want to delete.
3. Press GOLD and REPLACE.

In other words:

GOLD COMMAND CLEAR PASTE ENTER
Move the cursor to the first character.
SELECT
Move the cursor to the character following the last character.
GOLD REPLACE

5.7 Inserting a New Section of Text in Multiple Locations

You can use the SELECT, CUT, and PASTE functions to insert a new section of text in multiple locations.

1. Press SELECT.

2. Type in the text you want to insert.

   KED highlights the text as you type it by underlining it or displaying it in reverse video (reversing the text and background).

3. Press CUT.

   KED deletes the text you typed and stores it in a paste buffer.

4. Move the cursor to the proper location.

5. Press GOLD and PASTE.

   KED inserts the contents of the paste buffer (the text you typed) at the point where the cursor is located and moves the cursor to the first character following the inserted text.

6. Repeat steps 4 and 5 as many times as necessary.

In other words:

> SELECT `text-to-insert` CUT
>
> Move the cursor to the proper location.
>
> GOLD PASTE
>
> Move the cursor to another location.
>
> GOLD PASTE
>
> Move the cursor to another location.
>
> GOLD PASTE

Your text remains in the paste buffer until you press CUT (causing KED to store new text in that buffer) or until you enter a CLEAR PASTE command as described in Section 5.5.

5.8 Removing an Existing Section of Text and Inserting It in Another Location

To remove a section of text and insert it in another location, use the SELECT, CUT, and PASTE functions.

1. Select the section of text you want to cut.

   a. Move the cursor to the first character in that section.

   b. Press SELECT.

   c. Move the cursor to the character following the last character in that section.

      KED highlights your selection by underlining it or displaying it in reverse video (reversing the text and background).

2. Press CUT.

   KED deletes the text you selected and stores it in a paste buffer. If that buffer is too small to hold your selection, select and cut a smaller section of text (steps 1 and 2). After pasting that section in another location (step 3), select, cut, and paste the rest of the text.

3. Paste your selection in another location.

   a. Move the cursor to the other location.

   b. Press GOLD and PASTE.

      KED inserts your selection at the point where the cursor is located and moves the cursor to the first character following the inserted text.

In other words:

Move the cursor to the first character.
SELECT
Move the cursor to the character following the last character.
CUT
Move the cursor to the other location.
GOLD PASTE

You can press GOLD and PASTE again and again to insert your selection in multiple locations. Your selection remains in the paste buffer until you press CUT (causing KED to store new text in that buffer) or until you enter a CLEAR PASTE command as described in Section 5.5.

For large cut and paste operations, you can use an auxiliary file for temporary storage as described in Chapters 7 and 8.

Removing and Inserting Two or More Sections of Text

To remove two or more sections of text from different locations in your file, join them together, and insert them in another location:

1. Select and cut the first section.

   a. Move the cursor to the first character in that section.

   b. Press SELECT.

   c. Move the cursor to the character following the last character in that section.

   d. Press CUT.

   KED deletes that section and stores it in the paste buffer.

2. Select (do not cut) the next section by repeating steps a, b, and c.

3. Press APPEND.

   KED deletes that section and adds it to the end of the first section in the paste buffer.

4. Continue selecting and appending sections of text (by repeating steps 2 and 3) as many times as necessary.

   KED deletes each section you select and adds it to the end of the contents of the paste buffer.

5. Move the cursor to the point in your text where you want to insert the appended sections.

6. Press GOLD and PASTE.

   KED inserts the entire contents of the paste buffer (the appended sections) in your file.

In other words:

Move the cursor to the first character.
SELECT
Move the cursor to the character following the last character.
CUT
Select (do not cut) the next section.
APPEND
Repeat the selecting and appending operations.
Move the cursor to the point of insertion.
GOLD PASTE

5.9 Replacing One Section of Text with Another

To replace one section of text with another:

1. Select the replacement text and store it in the paste buffer.
   a. Move the cursor to the first character in the replacement text.
   b. Press SELECT.
   c. Move the cursor to the character following the last character in the replacement text.
   d. Press CUT.

      KED stores the replacement text in the paste buffer.
2. Select the text you want to delete.
   a. Move the cursor to the first character in the text you want to delete.
   b. Press SELECT.
   c. Move the cursor to the character following the last character in the text you want to delete.
3. Press GOLD and REPLACE.

   KED deletes the selected text and replaces it with the contents of the paste buffer, moving the cursor to the first character following the replacement text.

In other words:

Move the cursor to the first character in the replacement text.
SELECT
Move the cursor to the character following the last character in that text.
CUT
Move the cursor to the first character in the text to be deleted.
SELECT

Move the cursor to the character following the last character in that text.
GOLD REPLACE

If you use the FIND function to select the text to be deleted, KED stores that text in its word buffer. You can recover it by pressing GOLD and UNDELWORD.

If you use the SELECT function to select the text to be deleted, you cannot recover that text unless you use a backup copy of your input file.

5.10 Substituting a Character String for Multiple Occurrences of Another String

To substitute a character string for multiple occurrences of another character string:

1. Select or type the replacement string and store it in the paste buffer.

   To select the replacement string:

   a. Move the cursor to the first character in the string.

   b. Press SELECT.

   c. Move the cursor to the character following the last character in that string.

   d. Press CUT.

      KED stores the selected string in the paste buffer.

   To type a new replacement string:

   a. Press SELECT.

   b. Type the new character string.

   c. Press CUT.

      KED stores the new string in the paste buffer.

2. Specify the string you want to delete.

   a. Press GOLD and FIND.

   b. Type the character string you want to delete.

   c. Press ADVANCE or BACKUP.

      KED moves the cursor forward or backward to the first character of the string you want to delete.

3. Perform the substitutions.

   a. Press GOLD and SUBSTITUTE.

      KED substitutes the replacement string for the string to be deleted, searches for the next occurrence of the string to be deleted, and moves the cursor to the first character of that string.

   b. Repeat step a as many times as necessary.

You can use the GOLD number function described in Section 5.13 to repeat that step a certain number of times.

In other words:

Select the replacement string or type a selected replacement string.

CUT

Use the FIND function to select the string to be deleted and to move the cursor.

GOLD SUBSTITUTE GOLD SUBSTITUTE GOLD SUBSTITUTE

To use the SUBSTITUTE function for deleting multiple occurrences of a character string:

1. Empty the paste buffer as described in Section 5.5.
2. Specify the string you want to delete, using the FIND function.
3. Press GOLD and SUBSTITUTE as many times as necessary.

In other words:

GOLD COMMAND CLEAR PASTE ENTER

Use the FIND function to select the string to be deleted and to move the cursor.

GOLD SUBSTITUTE GOLD SUBSTITUTE GOLD SUBSTITUTE

5.11 Forcing the Case of Alphabetic Characters to Lower or Upper

You can force the case for alphabetic characters to be lower or upper, using the GOLD/L and GOLD/U keyboard keys.

Pressing GOLD/L forces to lowercase the character on which the cursor is resting or all characters within an active select range.

Pressing GOLD/U forces to uppercase the character on which the cursor is resting or all characters within an active select range.

5.12 Changing the Case of Alphabetic Characters

You can change the case of alphabetic characters from uppercase to lowercase or from lowercase to uppercase, using the GOLD and CHNGCASE keys.

Changing a Character

1. Move the cursor to the alphabetic character you want to change.
2. Press GOLD and CHNGCASE.

   KED deletes the character you selected and replaces it with the same character in the other (upper or lower) case.

Changing a Character String

1. Move the cursor to the first character in the string you want to change.
2. Press SELECT.

3. Move the cursor to the last character in that string.

   The character string you select can contain numeric and other nonalphabetic characters, but they will not be affected.

4. Press GOLD and CHNGCASE.

   KED deletes the alphabetic characters you selected and replaces them with the same characters in the other (upper or lower) case.

In other words:

> Move the cursor to the first character in the string to be changed.
> SELECT
> Move the cursor to the last character in that string.
> GOLD CHNGCASE

Using the FIND Function with Selection Operation

You can also use the FIND function in changing the case of an alphabetic character or character string.

To change the case of a character or an entire character string:

1. Enable the selection operation in the FIND function.

   a. Press GOLD and COMMAND.

   b. Type:

   ```
   SET [SEARCH] SELECT
   ```

   c. Press ENTER.

2. Press GOLD and FIND.

3. Type the character or character string.

4. Press ADVANCE or BACKUP.

5. Press GOLD and CHNGCASE.

   KED changes the case of the character or characters you selected with the FIND function.

In other words:

> GOLD COMMAND `SET SELECT` ENTER
> Use the FIND function to find the character string to be changed.
> Move the cursor to the last character in that part.
> GOLD CHNGCASE

To disable the selection operation in the FIND function:

> GOLD COMMAND `SET NOSELECT` ENTER

5.13 Repeating Editing Operations

You can repeat editing operations in three ways:

- If your terminal has the autorepeat feature, continue pressing (hold down) the keypad or keyboard key that will give you the operation you want.
- Create and use a KED macro (a sequence of functions and commands) as described in Chapter 6.
- Use the GOLD number function as described below.

You can use the GOLD number function in the following procedures for all functions except delete, CTRL/U, or SPECINS.

When the Operation Must Be Specified After the GOLD Number Function:

1. Press GOLD.
2. Use the numeric keys on your keyboard to type the number of times you want the operation to be performed. The maximum number is 32767.

   KED displays its `Repeat:` prompt and the number you typed.
3. Press one or more of the keypad or keyboard keys that will perform the operation.

   KED performs the operation the number of times you specified.

For example, to advance the cursor three sections:

ADVANCE GOLD 3 SECTION

You can cancel a repeat operation while KED is performing it by pressing CTRL/C. If KED is displaying its `Repeat:` prompt, press ENTER.

When Part of the Operation Must Be Specified Before the GOLD Number Function:

You can insert a character string a certain number of times by combining the SELECT, CUT, and PASTE functions (Section 5.7) with the GOLD number function. For example, if you want to insert the string `[]` 20 times:

1. Press SELECT.
2. Type:

   ```
   []
   ```
3. Press CUT.
4. Press GOLD.
5. Type:

   ```
   20
   ```
6. Press GOLD and PASTE.

   KED displays:

```
[][][][][][][][][][][][][][][][][][][][]
```

In other words:

SELECT [ ] CUT GOLD 20 GOLD PASTE

Other Useful Examples:

The procedures described above are also useful when you want to insert consecutive single characters. For example, if you want to insert a line of 50 asterisks as a border for a comment block in a program, the following procedure is the quickest way to produce that line:

1. Press GOLD.
2. Type the number 50 on your keyboard.
3. Press the Shift and * keys on your keyboard.

   KED displays:

   ```
   **************************************************
   ```

In other words:

GOLD 50 Shift/*

5.14 What Next?

If you did not complete the work session example in Appendix E to verify your installation, you should complete that example now, because it will help you review and summarize many of the operations you have learned so far.

After you complete the work session example, you should know enough about KED to make full use of the chapters remaining in this document.

- Chapter 6 tells you how to use KED macros and command files to perform many editing operations in a more efficient manner.
- Chapters 7 and 8 tell you how to be more efficient by using auxiliary files to move text from one file to another.
- Chapter 9 tells you how to begin your work session so you can recover all or most of the work you performed in the event of a system crash or similar malfunction.
- Chapter 10 gives you three very useful operations for program development.

Chapter 6
Using Macros and Command Files

Some editing operations require the repetition of two or more KED functions or commands. For example, the following sequence deletes one or more spaces at the end of text lines in a large file.

1. Press ADVANCE.
2. Press EOL to move the cursor to the end of a text line.
3. Press <X as many times as necessary to delete the spaces at the end of that line, while making sure that you do not accidentally delete any printing characters.
4. Repeat steps 2 and 3 as many times as necessary to delete the spaces at the end of all text lines in the file, while making sure that you do not accidentally delete any new-line terminators that precede blank lines.

The easiest and most accurate way to perform such tedious operations is to use a KED macro consisting of a sequence of functions or commands that can be repeated. For example, the following macro (described in Section 6.8.4) performs steps 1, 2, and 3 of the operation shown above and it does not accidentally delete any printing characters or new-line terminators.

GOLD FIND SPACE BAR RETURN ADVANCE
BACKUP WORD GOLD FIND SPACE BAR ADVANCE
GOLD DELEOL

Also, if that macro is preceded by the repeat function (GOLD `number-of-lines`), it performs step 4 of the operation shown above.

This chapter tells you how to:

- Create and store macros (Section 6.1)
- Execute macros (Section 6.2)
- Conserve space in the macro buffer (Section 6.3)
- Clear the macro buffer (Section 6.4)
- Create and execute macros in auxiliary files (Section 6.5)
- Create and use KED command files (Section 6.6)
- Use KED command files as work-session initialization files (Section 6.7)

6.1 Creating and Storing Macros

You can create and store one or more KED macros. In one work session, the total number of characters in the macros cannot exceed 512, unless you perform a customization to increase the macro buffer capacity.

To create and store a KED macro, you must begin its definition (Section 6.1.1) and then end its definition (Section 6.1.2).

6.1.1 Beginning the Definition of a Macro

Use the LEARN command to begin the definition of your macro.

1. Press GOLD and COMMAND.
2. Type:

   ```
   LEARN [identifier]
   ```

   `identifier` can be any single letter in the alphabet or the backslash (\) character.

   A single macro does not require an identifier and is, by default, assigned the backslash character. Any subsequent macros you create in the same work session (that you want to preserve) do require a unique identifier.

   Subsequent LEARN operations close previous LEARN operations. If a subsequent LEARN macro is given the same name as a previous LEARN macro, the subsequent replaces the previous. For example, if you do not identify a macro, subsequently identifying another macro with the backslash character will replace the unidentified macro.
3. Press ENTER or RETURN.

   If you press ENTER, KED returns to text insertion mode. If you press RETURN, KED remains in command mode.
4. Press the function keys and enter the commands in the order you want them executed.

   You cannot include the GOLD S or GOLD X functions, or the LEARN, EXIT, or QUIT commands in your macro. All other functions and commands are valid, up to a maximum of 512 characters. KED executes and displays the effect of each successful function and command as you enter it. If your macro definition encounters an error condition, you should stop the definition.

   KED places your sequence of functions and commands in a macro buffer. That sequence remains in the buffer until you enter a CLEAR command (Section 6.4).

   Finally, encountering an error condition does not terminate the LEARN macro creation. However, an error condition does terminate the execution of a LEARN macro. Therefore, you should ensure that an error encountered while creating a macro is avoided while executing the macro. The safest procedure is to restart the LEARN operation and avoid the condition that produced the error condition.

In other words:

> GOLD COMMAND `LEARN [identifier]` ENTER
> Press function keys and enter commands in the order you want them executed.

If your macro is too large for the buffer, KED clears that macro from the buffer and signals you by sounding a beeper or reversing the text and background on your terminal. If you press HELP, KED displays:

```
Macro buffer full
```

If your macro is too large for the buffer, you can reduce the size of your macro as described in Section 6.3, or you can create and use a KED command file as described in Section 6.6.

6.1.2 Ending the Definition of a Macro

Use the GOLD S or GOLD X function to end the definition of your macro.

1. Press GOLD.
2. Type:

   ```
   S
   ```

   or

   ```
   X
   ```

   KED ends the definition of your macro and marks it as the current macro.

   > If you type `S`, KED does not execute it.
   > If you type `X`, KED executes it.

In addition to storing macros in the macro buffer, you can store them in auxiliary files as described in Section 6.5.

6.2 Executing Macros

Regardless of how you created your macro or how it is marked, you can execute it once or a specified number of times. But if your macro encounters an error while executing, KED stops executing the macro when the macro encounters that error condition. Consequently, you should watch your screen when you first execute the macro, to make sure that KED can complete all the operations you defined.

To execute a macro marked as the current macro:

1. Press GOLD.
2. Type:

   ```
   X
   ```

   KED executes the current macro (one created previously and marked as the current macro).

The GOLD X function will execute any macro marked as the current macro, regardless of how or when it was created. In other words, you can use the GOLD X function to execute a macro that you executed previously with the GOLD M identifier function described below.

To execute any macro, whether or not it is marked as the current macro:

1. Press GOLD.
2. Type:

   `M identifier`

   `identifier` must be the letter you specified in a `LEARN identifier` command or the backslash (\) character for a macro you created with the `LEARN` command.

   KED executes the macro specified by your identifier and marks it as the current macro.

Later, you can use the GOLD X function described above to execute the same macro, provided it is still marked as the current macro.

If you want to execute a macro a certain number of times:

1. Press GOLD.
2. Type a number (on your keyboard) to specify the number of times you want to execute your macro.
3. Press GOLD and type:

   `X`

   or

   `M identifier`

   KED executes the macro the number of times you specified in step 2.

In other words:

GOLD `number` GOLD `X`
or
GOLD `number` GOLD `M identifier`

In addition to executing macros from the macro buffer, you can execute them from auxiliary files as described in Section 6.5.

When creating auxiliary files in a macro:
KED assumes that any auxiliary file that currently exists on your system can be overwritten and, therefore, does not prompt for permission to overwrite (replace) the file. KED does not return the prompt, `Auxiliary file exists. Replace?`, when an auxiliary file is created in a macro.

6.3 Conserving Space in the Macro Buffer

The macro buffer can hold up to 512 characters (some systems allow a larger number). Each keypad or keyboard key you type adds one or more characters to that buffer.

Command, Function, or Key	Characters (Bytes) Added to Buffer
Keypad key	3
Arrow keyboard key	3
Other keyboard key	1
KED command	9 plus 1 for each keyboard key typed
GOLD number	3 plus 1 for each digit typed

If your macro is too large for the buffer, KED signals you by sounding a beeper or reversing the text and background on your terminal, and clears that macro from the buffer. If you press HELP, KED displays:

```
Macro buffer full
```

The following suggestions will help you reduce the size of your macros and thus conserve space in the macro buffer.

- Use command abbreviations or equivalent functions and commands as described in Appendix C.
- If your macro finds a character string, specify the model string before you create the macro. Then, use the FINDNEXT function instead of the FIND function in the macro.
- Specify KED settings, such as screen width and page definitions, before you create the macro.
- If your macro inserts text in your file, store that text in the paste buffer before you create the macro. Then, use the PASTE function in the macro.
- Avoid using the ADVANCE and BACKUP functions in your macro by organizing it so that all operations proceed in one direction. Then, you can press ADVANCE or BACKUP (depending on the direction you want to go) before executing the macro.
- Use the repeat (GOLD number) function in your macro to repeat an operation three or more times.

In addition to reducing the size of your macros, you can clear one or more identified macros from the macro buffer as described in Section 6.4, or create and use one or more command files as described in Section 6.6.

6.4 Clearing the Macro Buffer

You can clear (delete) all macros or individual macros from the macro buffer.

To clear all macros:

1. Press GOLD and COMMAND.
2. Type:

   ```
   CLEAR ALL MACROS
   ```

3. Press ENTER.

   KED clears all macros from the macro buffer.

In other words:

GOLD COMMAND `CLEAR ALL` ENTER

To clear individual macros:

1. Press GOLD and COMMAND.
2. Type:

   ```
   CLEAR MACRO identifier
   ```

   `identifier` must be the letter you specified in a `LEARN identifier` command or the backslash (\) character for a macro you created with the `LEARN` command.

3. Press ENTER.

   KED clears the macro you identified and merely consolidates any other macros remaining in the buffer.

In other words:

GOLD COMMAND `CLEAR MACRO identifier` ENTER

6.5 Creating and Executing Macros in Auxiliary Files

You can create a macro in an auxiliary file and then execute it from the macro buffer or directly from the auxiliary file. Also, once a macro is in an auxiliary file, you can execute that macro from another macro.

6.5.1 Creating Macros in Auxiliary Files

The procedure you should use for creating your macro depends on the size of the macro and how it will be executed.

If you think your macro will not be too large for the macro buffer:
You can use the LEARN identifier command, GOLD S function, and SAVE identifier command to create your macro in an auxiliary file.

1. Create and store your macro in the macro buffer, using the LEARN identifier command and the GOLD S function as described in Section 6.1.

2. Use the SAVE identifier command to copy your macro to an auxiliary file.

   a. Press GOLD and COMMAND.

   b. Type:

      ```
      SAVE identifier [filespec]
      ```

      `identifier` must be the same identifier you used in creating and storing your macro (step 1). If the identifier is the backslash (\) character, you must include the file specification.

      If you do not type `filespec`, the file name will be your identifier and the file type will be KED (identifier.KED). If you type just the file name, the file type will be KED.

   c. Press ENTER.

      KED copies your macro to the auxiliary file you specified.

In other words:

> Create and store your macro in the macro buffer.
> GOLD COMMAND SAVE `identifier [filespec]` ENTER

If you think your macro will be too large for the macro buffer or if you plan to execute your macro from another macro:
Do not use the LEARN identifier command, GOLD S function, and SAVE identifier command to create your macro in an auxiliary file. Instead:

1. End your current work session.
2. Use KED in a second work session to create (type) your macro in an auxiliary file, as described in Section 6.6.

Although the auxiliary file can be any standard file specification in the form *filnam.typ*, the following KED defaults are more useful:

> The file name is the single letter identifier you assigned the file with the SAVE command.
> The file type is KED.

If you find that your macro is not too large for the macro buffer, you will not have to add a file specification to the LOAD identifier command. If you find that your macro is small enough to be executed from another macro (see example in Section 6.6), those defaults will consume less space in the macro buffer.

6.5.2 Executing Macros in Auxiliary Files

If the macro in your auxiliary file contains an error, KED will stop its execution when KED reaches that error. Consequently, you should watch your screen when you first execute the macro, to make sure that KED can complete all the operations you defined.

The procedure you use for executing your macro depends on the size of the auxiliary file.

If your auxiliary file is not too large for the macro buffer:
You can use the LOAD identifier command and the GOLD M identifier function to execute your macro.

1. Use the LOAD identifier command to copy the auxiliary file to the macro buffer.

   a. Press GOLD and COMMAND.

   b. Type:

   ```
   LOAD identifier [filespec]
   ```

   `identifier` can be any single letter you choose, as long as you are not using it to identify another macro. If the identifier is the backslash (\) character, you must include a file specification.

   If you do not type `filespec`, the file name will be your identifier and the file type will be KED (identifier.KED). If you type just the file name, the file type will be KED.

   c. Press ENTER.

   KED copies the auxiliary file you specified to the macro buffer.

2. Execute your macro, using the GOLD M identifier function as described in Section 6.2.

   `identifier` must be the same identifier you used with the LOAD command to copy the auxiliary file to the macro buffer (step 1b).

In other words:

```
GOLD COMMAND LOAD identifier [filespec] ENTER
GOLD M identifier
```

If your auxiliary file is too large for the macro buffer:
You can use the @filespec command to execute your macro.

1. Press GOLD and COMMAND.

2. Type:

   ```
   @filespec
   ```

   `filespec` is the file specification you want to execute (default file type is KED)

3. Press ENTER.

   KED executes your macro directly from the file you specified.

In other words:

```
GOLD COMMAND @filespec ENTER
```

6.6 Creating and Using KED Command Files

You can use KED in another work session to create an auxiliary file containing KED commands that you would normally type in response to the `Command:` prompt. In that file:

- You cannot use the LEARN command to define a macro.
- An exclamation point (!) can be used to indicate that the rest of a line is a comment, not a command. If you plan to copy the file to the macro buffer, keep your comments short.
- When the macro is executed, blank lines are skipped by KED when it is in the command mode.
- The CHANGE command, which returns KED to the text editing mode, can precede text that you want to insert directly into your output file.
- KED assumes that any auxiliary file that currently exists on your system can be overwritten and, therefore, does not prompt for permission to overwrite (replace) the file. KED does not return the prompt, `Auxiliary file exists. Replace?`, when run from command files.

The following is an example of a command file.

```
! We are in command mode
SET WRAP 30     ! Set right margin

! Change to Insert mode
CHANGE
Use the R command if the input file is on your system volume.

If that file is on another volume, use the RUN command.
```

That command file sets the length of text lines to 30 columns and then inserts the text as shown below.

```
Use the R command if the input
file is on your system volume.

If that file is on another
volume, use the RUN command.
```

After you create your command file:

- If it is not too large for the macro buffer, you can load it as a macro with the LOAD identifier command and then execute it as a macro with the GOLD M identifier function, as described in Section 6.5.2.
- If it is too large for the macro buffer or if you plan to use it once, you can execute it with the @filespec command, as described in Section 6.5.2.

If the macro in your command file contains an error, KED will stop its execution when KED reaches that error. Consequently, you should watch your screen when you first execute the macro, to make sure that KED can complete all the operations you defined.

6.7 Using KED Command Files as Work-session Initialization Files

Work-session initialization files are KED command files (Section 6.6). Initialization files can preset KED to give you certain operations and characteristics in every work session or let you complete a work session without typing any keypad or keyboard keys.

Before beginning a work session, KED looks for initialization file KEDINI.KED. You can use KED to create KEDINI.KED if you want to use a default initialization file. If KEDINI.KED exists, KED executes it. If it does not exist, KED does not report an error.

If you want KED to execute a different initialization file before beginning your work session, you can use KED to create the file and then use the following command to begin your work session:

EDIT/COMMAND:init-filespec filespec RET

init-filespec   is the specification for another initialization file.

filespec   is the specification for your input or output file.

If the other initialization file exists, KED executes it instead of KEDINI.KED. If the other initialization file does not exist, KED displays:

```
?KED-W-Command file not found - Continue (Y,N)?
```

Type Y and press RETURN to begin your work session without executing any initialization file. Type N and press RETURN to return to your system prompt.

If you want to suppress that query and any others KED could generate, use the following command to begin your work session:

EDIT/NOQUERY filespec RET

or

EDIT/COMMAND:init-filespec/NOQUERY filespec RET

If KEDINI.KED exists and you do not want KED to execute any initialization file, use the following command to begin your work session.

EDIT/NOCOMMAND filespec RET

The following is an example of an initialization file that presets KED to give you certain operations and characteristics in every work session.

```
! Set Session Characteristics
SET QUIET                       ! Reverse screen to indicate errors
SET SECTION 10                  ! Define section as 10 lines
SET NOHELP                      ! Disable full help screens
SET SEARCH NOSELECT             ! Disable selection default
```

The following is an example of an initialization file that lets you complete a work session without typing any keypad or keyboard keys.

```
! We start in command mode
! So we want to change to insert mode to
! Insert Page Head
CHANGE
For Internal Use Only
! We now return to command mode
! by including the COMMAND escape equivalent
πOPπOw
! And
EXIT
```

KED stops its execution if the macro in your initialization file contains an error or encounters an error condition. Consequently, you should watch your screen when you first execute the macro, to make sure that KED can complete all the operations you defined.

6.8 Examples

This section contains some examples of KED macros that you may find useful or want to modify to suit your needs.

Each example has three parts.

1. The commands and functions you use before entering the LEARN command
2. The LEARN command and the combination of commands and functions you want KED to store as a macro
3. The commands and functions that normally follow the GOLD S or GOLD X function you must use to end the macro

Each example is described as a series of numbered steps with at least one command or function in each step.

6.8.1 Decreasing the Size of a BASIC–PLUS/RT Program by Removing the Text in REM Statements

You can decrease the size of a BASIC–PLUS/RT program by removing the comments. But you should remove only the text, because transfer statements, such as the GO TO statement, may specify the line number of a REM statement.

You can remove the text in a series of REM statements by executing the following macro. It assumes that the program does not contain compound statements in the form `10 REM This is a comment \ PRINT X$`.

1. Set KED for exact search matching.

   ```
   GOLD  COMMAND  SET SEARCH EXACT  ENTER
   ```

2. Specify that KED will place the cursor after the target in a successful search.

   ```
   GOLD  COMMAND  SET SEARCH END  ENTER
   ```

3. Specify the string REM as the search target.

   ```
   GOLD  FIND  REM  ADVANCE
   ```

4. Start storing a sequence of functions (the macro).

   GOLD COMMAND LEARN ENTER

5. Delete the text in each REM statement but leave the line number and the word REM.

   GOLD DELEOL FINDNEXT

6. End the sequence.

   GOLD S

7. Execute the macro.

   ADVANCE GOLD X

   or

   BACKUP GOLD X

6.8.2 Extracting Addresses That Contain a Certain ZIP Code

When you have a mailing list that uses the same format for each address, you may be able to use a macro to extract addresses that contain a certain ZIP code.

The following macro assumes that your mailing list has:

A blank line before the first line of each address
No 5-digit number except the ZIP code in any address

Since a blank line precedes each address, each address is preceded by two new-line terminators. The macro uses that format to guarantee that addresses with any number of lines will be extracted.

1. Define a page as the text between markers composed of two new-line terminators.

   GOLD COMMAND SET PAGE "RETURN RETURN" ENTER

2. Open auxiliary output file ADDOUT.LST for the addresses to be extracted.

   GOLD COMMAND OPEN OUTPUT ADDOUT.LST ENTER

3. Specify ZIP code 99999 as the search target.

   GOLD FIND 99999 ADVANCE

   or

   GOLD FIND 99999 BACKUP

4. Start storing a sequence of commands and functions (the macro).

   GOLD COMMAND LEARN ENTER

5. Find the next address that contains the ZIP code you specified.

   ADVANCE FINDNEXT

6. Move the cursor back to the beginning of that address.

   BACKUP PAGE

7. Write the address to the auxiliary output file.

   GOLD COMMAND WRITE 1 PAGE ENTER

8. End the sequence.

   GOLD S

9. Execute the macro.

   GOLD X

You can use that macro to extract addresses for more than one ZIP code by using the FIND function again to specify a different ZIP code.

6.8.3 Moving the Cursor to the End of a Word

The WORD function moves the cursor to the first character in a word, but you may want to change the endings of several different words. For example, you may want to change some words from the singular to the plural.

In normal text, most of the words that KED detects end with one or more spaces. This macro uses that format to move the cursor to the first space that follows the printing characters in a word.

1. Specify a space as a search target.

   GOLD FIND SPACE BAR ADVANCE

2. Start storing a sequence of functions (the macro).

   GOLD COMMAND LEARN ENTER

3. Move the cursor to the beginning of a word and then to the first space after that word.

   WORD FINDNEXT

4. End the sequence and execute the macro.

   GOLD X

That macro moves the cursor to the end of a word only when the word has trailing spaces. The macro skips words that end with line terminators and horizontal-tab characters.

6.8.4 Deleting Spaces from the End of a Text Line

When you create a text file, each line in the file may have several spaces preceding the line terminator. The following macro assumes that a new-line terminator is at the end of each line and deletes any spaces that are between the last printing character and the terminator.

1. Start storing a sequence of functions (the macro).

   GOLD COMMAND LEARN ENTER

2. Search for a text line that has one or more trailing spaces.

   GOLD FIND SPACE BAR RETURN ADVANCE

3. Move the cursor to the first trailing space in the line.

   BACKUP WORD GOLD FIND SPACE BAR ADVANCE

4. Delete all trailing spaces.

   GOLD DELEOL

5. End the sequence and execute the macro.

   GOLD X

6.8.5 Reformatting Paragraphs

The FILL function or command reformats selected text lines. When KED finishes processing the selected lines, no new line reaches beyond the right margin you are using.

When you want to reformat text lines that are in separate paragraphs, you must select each paragraph and then use the FILL function or command.

The following macro reformats paragraphs when they are separated by two new-line terminators that are followed by five spaces at the beginning of the first line. The macro assumes you will advance, not back up, through your file from paragraph to paragraph.

1. Specify two new-line terminators as the search target.

   GOLD FIND RETURN RETURN ADVANCE

2. Start storing a sequence of commands and functions (the macro).

   GOLD COMMAND LEARN ENTER

3. Move the cursor to the first printing character in the paragraph.

   GOLD 5 →

4. Select the text lines that are in the paragraph.

   SELECT FINDNEXT

5. Reformat the text lines in the paragraph.

   GOLD FILL

   or

   GOLD COMMAND FILL ENTER

6. End the sequence.

   GOLD S

7. Execute the macro.

   GOLD X

6.9 Useful Command Files

You can create the following command file, PRINT.KED, to use during a work session to print selected text on a printer.

```
! PRINT.KED
!
OPEN OUTPUT LP:
WRITE SELECT
CLOSE
```

Issue the @PRINT command to execute that file.

You can create the following command file, FORM.KED, to use as an initialization file to insert an empty form (FORM.TXT) and then set some KED characteristics so your new form can be completed properly.

```
! FORM.KED
!
OPEN INPUT SY:FORM.TXT
INCLUDE REST
SET SCREEN 132
SET WRAP 120
SET TABS 10
```

Issue the following command to execute that file:

```
EDIT/CREATE/COMMAND:FORM.KED filespec RET
```

where `filespec` is your new form.

You can create the following command file, TRANSF.KED, to use as an initialization file to display a transfer address from a link map and then end your work session without producing an output file.

```
   ! TRANSF.KED
   ! We are in command mode by default
   ! Change to insert mode
   CHANGE
   ! And
   ! Include escape equivalents for FIND Transfer ADVANCE SELECT BLINE
   πOPπORTransferπOtπOnπOp
   ! We now return to command mode
   ! by including the COMMAND escape equivalent
   πOPπOw
   ! And
   OPEN OUTPUT TT:
   WRITE SELECT
   CLOSE
   QUIT
```

Issue the following command to execute that file:

```
EDIT/COMMAND:TRANSF.KED filespec RET
```

where `filespec` is the link map.

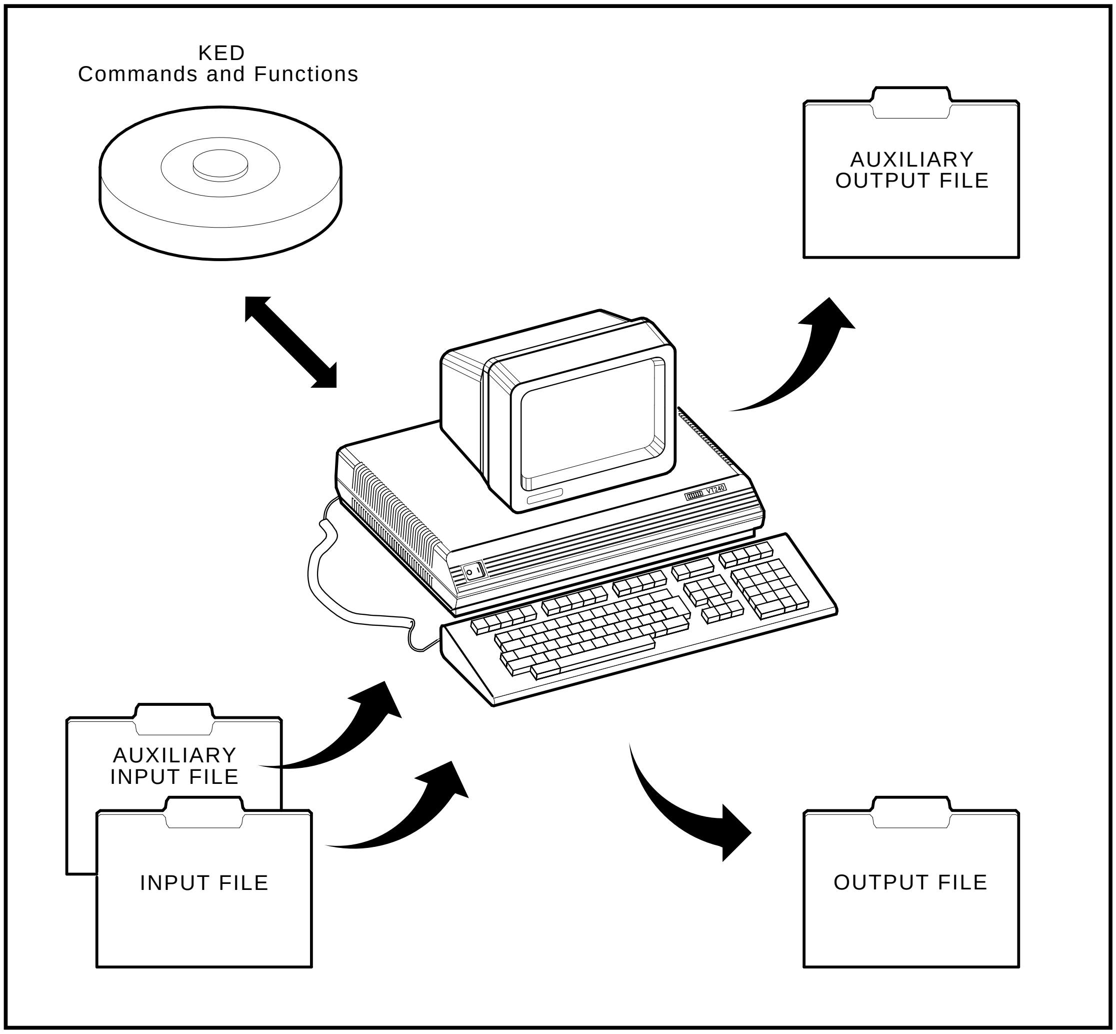
See Section 10.1 for a list of other keypad operations performed by escape equivalent characters like those shown above.

Chapter 7

Copying Text to Auxiliary Output Files

For large cut and paste operations, you can make a copy of any part of the file you are editing and store it in a separate (auxiliary) output file. Then, you can insert all or part of the contents of that file (then called an auxiliary input file) in another part of the file you are editing.

If you want to split your file into two files, after creating and closing the auxiliary output file, simply delete the part of your file that you stored in the auxiliary file.



MLO-003504

Although KED prevents you from changing a file while you are inspecting it, you can open an auxiliary output file, copy parts of the file you are inspecting to the auxiliary file, and close the auxiliary file without changing the file you are inspecting. Then you can edit the auxiliary file for another purpose and print it.

This chapter tells you how to:

- Open (create) an auxiliary output file that will contain the part you want to copy (Section 7.1)
- Select the part you want to copy and move that part to the auxiliary file (Section 7.2)
- Append more text to the auxiliary file (Section 7.3)
- Close the auxiliary file (Section 7.4)
- Purge (delete) the auxiliary file (Section 7.5)

Chapter 8 tells you how to insert all or part of the contents of your auxiliary file or another file in the file you are editing.

7.1 Opening (Creating) an Output File

When you open a file that does not exist, you are creating a new file.

You can open an auxiliary output file while you are creating, editing, or inspecting a file.

To open an auxiliary output file:

1. Press GOLD and COMMAND.
2. Type:

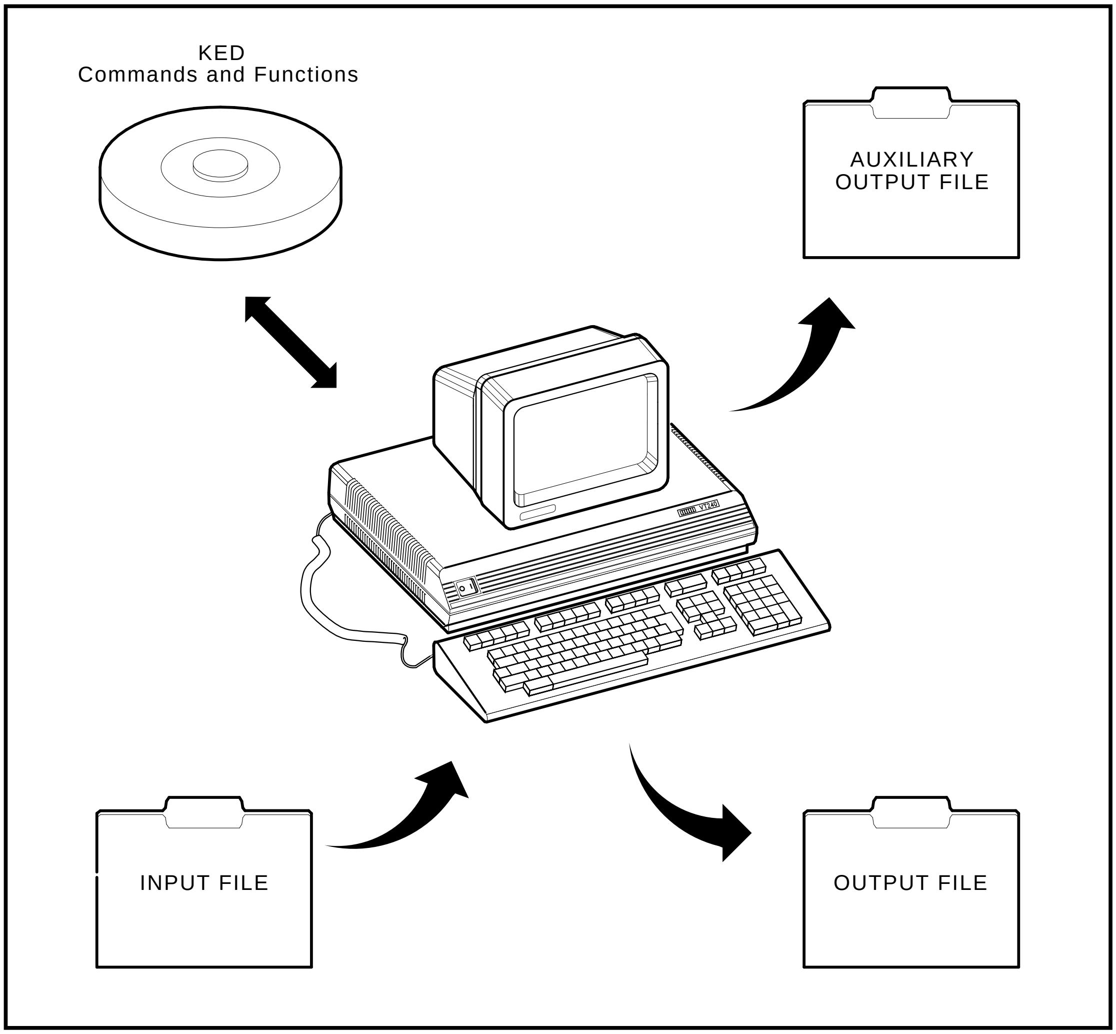
   ```
   [OPEN] OUTPUT filespec
   ```

   `filespec` is the file specification of a new file that will receive your text. Ensure that file specification does not duplicate another file specification. There is no default file type for auxiliary output files.
3. Press ENTER.

   KED creates and opens the file you specified.

In other words:

GOLD COMMAND `OUTPUT filespec` ENTER



MLO-003505

If you specify a file that already exists, KED signals you by sounding a beeper or reversing the text and background on your terminal and displays:

```
Auxiliary output file exists - Replace (Y,N)?
```

Type Y and press RETURN to delete the existing file and create a new output file with the same file specification. Type N and press RETURN to cancel your OPEN OUTPUT command.

You can use only one auxiliary output file at a time. If you issue an OPEN OUTPUT command to open a second auxiliary file without closing the first file, KED closes the first file (saving its contents in a separate file) and then opens the second auxiliary file. The first file is closed even when the OPEN OUTPUT operation for a second file is aborted; when, for example, you determine that a file specification for the second file already exists and you decide to stop the operation.

7.2 Selecting Text to Copy and Moving It to the File

You can use one of four WRITE commands to select the text you want to copy and to move that text to the auxiliary output file:

WRITE SELECT (Section 7.2.1)
WRITE REST (Section 7.2.2)
WRITE number LINES (Section 7.2.3)
WRITE number PAGES (Section 7.2.4)

In response to each command, KED starts with the character the cursor is on and copies all or part of the file you are inspecting, editing, or creating to the auxiliary output file. The text that KED copies is to the right of the cursor (and on lower lines). If you copied text with an earlier WRITE command, KED adds the new text at the end of the auxiliary output file.

7.2.1 WRITE SELECT Command

The WRITE SELECT command lets you use the SELECT function to select a section of text.

1. Move the cursor to the first character in the text you want to copy.
2. Press SELECT.
3. Move the cursor to the character following the last character in the text you want to copy.
4. Press GOLD and COMMAND.
5. Type:

   ```
   WRITE SELECT
   ```

6. Press ENTER.

   KED copies the selected text to the auxiliary file. Your original text remains intact.

In other words:

Select the text to be copied.
GOLD COMMAND `WRITE SELECT` ENTER

7.2.2 WRITE REST Command

The WRITE REST command selects all the text from the cursor to the end-of-file marker.

1. Move the cursor to the first character in the text you want to copy.
2. Press GOLD and COMMAND.
3. Type:

   ```
   WRITE REST
   ```

4. Press ENTER.

KED copies the rest of the text to the auxiliary file. Your original text remains intact.

In other words:

Move the cursor to the first character in the text to be copied.
GOLD COMMAND `WRITE REST` ENTER

7.2.3 WRITE number LINES Command

The WRITE number LINES command lets you specify the number of text lines you want to select.

1. Move the cursor to the first character in the text you want to copy.
2. Press GOLD and COMMAND.
3. Type:

   ```
   WRITE number [LINES]
   ```

   `number` can be from 1 to 32767.
4. Press ENTER.

   KED copies the selected text to the auxiliary file. Your original text remains intact.

In other words:

Move the cursor to the first character in the text to be copied.
GOLD COMMAND `WRITE number` ENTER

7.2.4 WRITE number PAGES Command

The WRITE number PAGES command lets you specify the number of pages you want to select. It uses the default definition of a page (all the text between form-feed characters) or the definition you created with the SET ENTITY PAGE command.

1. Move the cursor to the first character in the text you want to copy.
2. Press GOLD and COMMAND.
3. Type:

   ```
   WRITE number PAGES
   ```

   `number` can be from 1 to 32767.
4. Press ENTER.

   KED copies the selected text to the auxiliary file. Your original text remains intact.

In other words:

Move the cursor to the first character in the text to be copied.
GOLD COMMAND `WRITE number PAGES` ENTER

7.3 Appending Text to the File

As long as your auxiliary output file remains open, you can select and copy as many sections of text as you like from anywhere in the file you are editing.

Each time you enter a WRITE command (Section 7.2), KED copies the text you selected and adds it to the end of the contents of your auxiliary file.

Once closed, however, that file cannot be reopened as an auxiliary output file. If you have more text to copy, you must open a second auxiliary output file (Section 7.1).

7.4 Closing the File

When you finish copying text to your auxiliary output file, you can close it by opening a second auxiliary output file (Section 7.1) or by issuing a CLOSE command.

1. Press GOLD and COMMAND.
2. Type:

   ```
   CLOSE
   ```

3. Press ENTER.

   KED closes your auxiliary file, saving its contents.

In other words:

GOLD COMMAND `(CLOSE)` ENTER

Once you close a file, you can no longer use it as an auxiliary output file, but you can use it as an auxiliary input file as described in Chapter 8.

7.5 Purging (Deleting) the File

If you decide you do not want to save your current (open) auxiliary output file:

1. Press GOLD and COMMAND.
2. Type:

   ```
   PURGE
   ```

3. Press ENTER.

   KED deletes the auxiliary file and cancels your last OPEN OUTPUT command.

In other words:

GOLD COMMAND `PURGE` ENTER

If you want to create (open) another auxiliary output file, you must issue another OPEN OUTPUT command.

7.6 Example

This section summarizes most of the information in this chapter by providing an example of all the steps necessary for copying text to an auxiliary output file. Each step has an instruction to complete and the keys you must type to complete the instruction.

1. Begin your work session to edit file WEEK.TXT.

   ```
   EDIT WEEK.TXT  RET
   ```

2. Use the OPEN OUTPUT filespec command to open auxiliary output file AUXOUT.TXT.

   ```
   GOLD  COMMAND  OPEN OUTPUT AUXOUT.TXT  RET
   ```

3. Use the WRITE number LINES command to copy the first two lines of WEEK.TXT to AUXOUT.TXT.

   ```
   WRITE 2 LINES  RET
   ```

4. Use the CLOSE command to close and save the contents of AUXOUT.TXT.

   ```
   CLOSE  RET
   ```

5. End your work session with the EXIT or QUIT command.

   ```
   EXIT  ENTER
   ```

   or

   ```
   QUIT  ENTER
   ```

Now you are ready to insert the contents of AUXOUT.TXT in file WEEK2.TXT, as described in the first example at the end of Chapter 8.

Chapter 8

Inserting Text from Auxiliary Input Files

You can specify a closed auxiliary output file or any other text file as an auxiliary input file and then insert all or part of its contents in the file you are editing.

You can use an auxiliary input file and an auxiliary output file at the same time, combining text from the auxiliary input file with your input file and copying parts of the file you are editing to an auxiliary output file. You can use only one auxiliary input file and one auxiliary output file at a time, but by changing them during an editing session, you can combine and split several files in many ways.

Chapter 7 tells you how to create an auxiliary output file. This chapter tells you how to:

- Open an auxiliary input file (Section 8.1)
- Skip to the line or page in the auxiliary file where you want to begin selecting the section to be inserted (Section 8.2)
- Insert all or part of the text from the auxiliary file into the file you are editing (Section 8.3)
- Close the auxiliary file (Section 8.4)

8.1 Opening an Input File

You can open a closed auxiliary output file (Chapter 7) or any other text file as an auxiliary input file while you are creating or editing a file. You cannot open an auxiliary input file while you are inspecting a file.

To open an auxiliary input file:

1. Press GOLD and COMMAND.
2. Type:

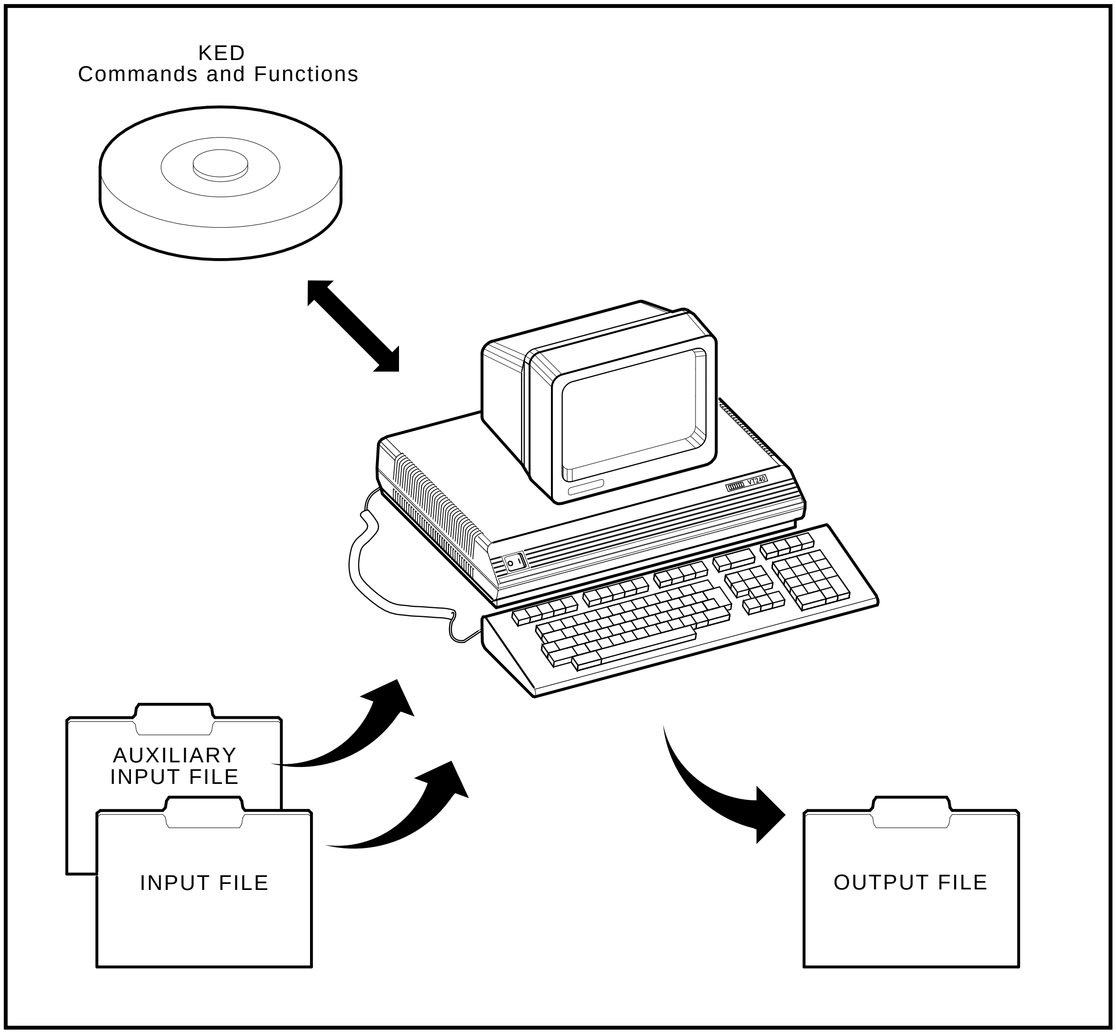
   ```
   [OPEN] INPUT filespec
   ```

   `filespec` is the file specification of an existing text file.
3. Press ENTER

   KED opens the file you specified.

In other words:

GOLD COMMAND `INPUT filespec` ENTER



MLO-003506

You can use only one auxiliary input file at a time. If you issue an OPEN INPUT command to open a second auxiliary file without closing the first input file, KED closes the first file and then opens the second auxiliary input file. The first file is closed even when the OPEN INPUT operation for a second file is aborted; when, for example, you learn that the specified file does not exist and you decide to stop the operation.

8.2 Skipping Text in the File

By default, KED starts at the beginning of your auxiliary input file. If the text you want to insert is somewhere else, you can use the SKIP number LINES or SKIP number PAGES command to specify another starting point in that file.

KED can skip lines and pages in one direction only—forward.

- If you skip past the point where you want to begin selecting text, you can close the input file as described in Section 8.4, and then reopen it as described in Section 8.1.
- If you skip to the end of the input file, KED closes it.

  If you try to skip again, KED signals you by sounding a beeper or reversing the text and background on your terminal. If you press HELP, KED displays:

  ```
  Auxiliary input file not open
  ```

8.2.1 SKIP number LINES Command

To skip one or more text lines in the input file:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SKIP number [LINES]
   ```

   `number` can be from 1 to 32767. You do not have to type `LINES`. If you type it, do not type the square brackets.
3. Press ENTER.

   Beginning at its current location in the input file, KED skips the number of lines you specified.

Type:

```
GOLD COMMAND SKIP number [LINES] ENTER
```

For example, to advance five lines:

```
GOLD COMMAND SKIP 5 LINES ENTER
```

8.2.2 SKIP number PAGES Command

To skip one or more pages in the input file:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SKIP number PAGES
   ```

   `number` can be from 1 to 32767. You must type `PAGES`.
3. Press ENTER.

   Beginning at its current location in the input file, KED skips the number of pages you specified.

In other words:

```
GOLD COMMAND SKIP number PAGES ENTER
```

For example, to advance five pages:

```
GOLD COMMAND SKIP 5 PAGES ENTER
```

In skipping pages, KED uses its default definition of a page (all the text between form-feed characters) or the definition you created with the SET ENTITY PAGE command. Be sure that the current page definition matches the way pages are marked in the input file.

8.3 Inserting Text from the File

The text you insert from an open auxiliary input file remains unchanged in that file.

To insert text from an auxiliary input file:

1. If necessary, enter a SKIP command as described in Section 8.2, to specify the point in the input file where you want to begin selecting text.
2. As shown below, enter an INCLUDE REST, INCLUDE number LINES, or INCLUDE number PAGES command to specify the number of lines or pages of text in the input file that you want to insert.

8.3.1 INCLUDE REST Command

To insert all the text in the input file, starting at the point where the cursor is located:

1. Press GOLD and COMMAND.
2. Type:

   ```
   INCLUDE REST
   ```
3. Press ENTER.

   KED inserts all the text at the point where the cursor is located, moves the cursor to the end of the insertion, and closes the input file.

In other words:

GOLD COMMAND `INCLUDE REST` ENTER

8.3.2 INCLUDE number LINES Command

To insert one or more lines of text from the input file, starting at the point where KED is located:

1. Press GOLD and COMMAND.
2. Type:

   ```
   INCLUDE number [LINES]
   ```

   `number` can be from 1 to 32767. You do not have to type `LINES`. If you type it, do not type the square brackets.
3. Press ENTER.

   KED inserts the selected text at the point where the cursor is located.

In other words:

GOLD COMMAND `INCLUDE number [LINES]` ENTER

8.3.3 INCLUDE number PAGES Command

To insert one or more pages of text from the input file, starting at the point where KED is located:

1. Press GOLD and COMMAND.
2. Type:

   ```
   INCLUDE number PAGES
   ```

   `number` can be from 1 to 32767. You must type `PAGES`.
3. Press ENTER.

   KED inserts the selected text at the point where the cursor is located.

In other words:

GOLD COMMAND `INCLUDE number PAGES` ENTER

In selecting pages, KED uses its default definition of a page (all the text between form-feed characters) or the definition you created with the SET ENTITY PAGE command. Be sure that the current page definition matches the way pages are marked in the input file.

8.4 Closing the File

KED closes the auxiliary input file when you end your work session. If you open a second input file while one is open, KED closes the first and opens the second input file.

In some situations you may want to close the input file. For example, if you skip beyond the point where you want to begin selecting text:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SKIP REST
   ```
3. Press ENTER.

   KED moves from its current location to the end of your input file and then closes it.

In other words:

GOLD COMMAND `SKIP REST` ENTER

8.5 Examples

This section summarizes most of the information in this chapter by providing examples of the steps necessary for copying text from an auxiliary input file to a file you are creating or editing.

Each step has two parts.

1. An instruction to complete
2. The keys you must type to complete the instruction

The keys you must type are presented in a horizontal line, and the strings of letters and numbers you must type are not enclosed by boxes.

This section provides examples of:

Inserting text from an auxiliary input file
Using auxiliary output and input files to move a large section of text

8.5.1 Inserting Text from an Auxiliary Input File

In the example at the end of Chapter 7, you created auxiliary output file AUXOUT.TXT. In the following procedure, you will use that file as an auxiliary input file and insert its contents in file WEEK2.TXT.

1. Begin your work session to create or edit WEEK2.TXT.

   ```
   EDIT WEEK2.TXT/CREATE  RET
   ```

   or

   ```
   EDIT WEEK2.TXT  RET
   ```

2. Use the OPEN INPUT filespec command to open AUXOUT.TXT as an input file.

   ```
   GOLD  COMMAND  OPEN INPUT AUXOUT.TXT  RET
   ```

3. Use the INCLUDE REST command to copy the contents of AUXOUT.TXT to the top of WEEK2.TXT and to close AUXOUT.TXT.

   ```
   INCLUDE REST  RET
   ```

4. End your work session with the EXIT command.

   ```
   EXIT  ENTER
   ```

8.5.2 Using Auxiliary Output and Input Files to Move a Large Section of Text

The KED paste buffer is usually large enough to let you move text around within your files. You may, however, want to move larger amounts of text than the paste buffer can hold. In that case, the general procedure for using auxiliary output and input files to move a large section of text is as follows:

1. Open an auxiliary (output) file.
2. Copy the section of text from your file to the auxiliary file, and close the auxiliary file.
3. Delete the section of text from your file.
4. Move the cursor to where you want the section of text to reappear in your file.
5. Open the auxiliary (input) file and copy the section of text from the auxiliary file to your file.

The following specific procedure for a particular file describes one method you can use. This example assumes that:

- Page numbers in your file appear at the bottom of each page and are enclosed by hyphens (for example, -7-).
- The pages are separated by the marker NEXTPAGE.
- The section you want to move includes all of pages 8, 9, and 10.
- You want to move those pages 150 lines backward in your file.

1. Open auxiliary output file AUX.TXT.

   ```
   GOLD  COMMAND  OPEN OUTPUT AUX.TXT  RET
   ```

2. Define a page in terms of the marker that separates your pages.

   ```
   SET PAGE "NEXTPAGE"  ENTER
   ```

3. Move the cursor to the beginning of the section of text you want to move.

   a. Search forward for the string -7-.

      ```
      GOLD  FIND  -7-  ADVANCE
      ```

   b. Move to the beginning of page 8.

      ```
      PAGE
      ```

4. Copy pages 8, 9, and 10 to the auxiliary output file.

   ```
   GOLD  COMMAND  WRITE 3 PAGES  RET
   ```

5. Close the auxiliary output file.

   ```
   CLOSE  RET
   ```

6. Delete the section of text you copied to the auxiliary output file.

   a. Clear the paste buffer.

      ```
      CLEAR PASTE  ENTER
      ```

   b. Select the 3-page section of text to be deleted.

      ```
      SELECT  BACKUP  PAGE  PAGE  PAGE
      ```

   c. Delete that section of text.

      ```
      GOLD  REPLACE
      ```

7. Move the cursor to where you want the section of text to reappear.

   ```
   BACKUP  GOLD  150  BLINE
   ```

8. Open the auxiliary output file as an input file.

   ```
   GOLD  COMMAND  OPEN INPUT AUX.TXT  RET
   ```

9. Copy the contents of the auxiliary file to your file.

   ```
   INCLUDE REST  ENTER
   ```

Chapter 9

Using Journal Files to Protect Work Sessions

If you begin your work session as described in Sections 2.1.1, 2.1.2, or 2.1.3 and your work session ends abnormally due to a power failure, user error, or system malfunction, you will lose all the work you performed. But, if you begin your work session as described below, you can recover all or most of the work you performed and then continue with your work session.

9.1 Beginning Your Work Session

A command line for beginning your work session with journal file protection has the general format:

EDIT/JOURNAL filespec[/option]

/JOURNAL can be abbreviated to /JOUR.

filespec is an output file specification if you want to create a new file or an input file specification if you want to edit an existing file.

/option can be any option you would normally include in beginning the work session, except /INSPECT.

That format produces a journal file, which will contain all or most of your work (the text you type and the changes you make). The journal file will have the file name you specify for `filespec`, but its file type will be JOU.

For example:

```
.EDIT/JOURNAL WEEK.TXT/CREATE RET
```

KED creates journal file WEEK.JOU, which will contain all or most of your work.

In some cases, you may want to use a different file specification for the journal file. A command line to do that has the general format:

EDIT/JOURNAL:jou-filespec filespec[/option]

jou-filespec is the different file specification you want to use for the journal file. You can type the file name, or you can type the file name and file type. If you type just the file name, the file type will be JOU.

For example, the following command line creates journal file RETAIN.JOU:

```
.EDIT/JOURNAL:RETAIN WEEK.TXT RET
```

Once you create a journal file, if your work session is terminated by an error or malfunction, you can use the /RECOVER option, as described in Section 9.2, to recover all or most of the work you performed.

9.2 Recovering the Work You Performed

A command line for beginning another work session and recovering all or most of the work you performed has the general format:

EDIT/RECOVER filespec[/option]

If you used a different file specification for the journal file, the general format is:

EDIT/RECOVER:jou-filespec filespec[/option]

/RECOVER can be abbreviated to /REC.

filespec is an output file specification if you created a new file or an input file specification if you edited an existing file.

jou-filespec is the file specification you used for a different journal file.

Either format uses the journal file to produce an output file that will contain all or most of the work you performed. The recovery operation starts at the beginning of a work session, before KED can execute an initialization file and before you can perform any additional work.

For example, type:

```
.EDIT/RECOVER WEEK.TXT  RET
```

or

```
.EDIT/RECOVER:RETAIN WEEK.TXT  RET
```

KED begins another work session and uses the journal file (WEEK.JOU or RETAIN.JOU) to recover all or most of the work you performed. You can then continue the work session or end it with an EXIT command.

A recovery operation does not automatically stop on an error, so you should watch your screen and check for any missing changes as the operation is being performed. The recovery operation reperforms the editing session and displays the session as it progresses. You can press HOLD SCREEN or CTRL/S to interrupt the recovery operation if you want to examine some part of the recovery more closely. Pressing again HOLD SCREEN or pressing CTRL/Q resumes the recovery operation.

You can terminate the recovery operation, if necessary, by twice pressing CTRL/C. Terminating recovery leaves you in the editor at that point in the file, so you can begin editing. If you terminated recovery because you noticed some correctable error in the work session you are recovering, you can issue the QUIT command, edit the journal file to correct the error, and restart the recovery operation.

Also, before you start a recovery operation:

- Make sure the conditions that existed during your previous work session still exist.

  Did the command line that began your previous work session execute an initialization file? If so, your command line for beginning the recovery operation should execute the same file.

During your previous work session, did you copy text to any auxiliary files? If so, those files should not contain any required text that cannot be replaced.

Did you also insert text or execute macros from any auxiliary files? If so, those files must exist in their previous condition.

- If the error or malfunction occurred while you were creating a new file, an output file was not produced. Therefore, you should use the following command line to begin the recovery operation.

  ```
  .EDIT/CREATE/RECOVER output-filespec  RET
  ```

- In the rare circumstance that you were editing an existing file and you think that the error or malfunction occurred after the input file type was renamed to BAK but before the output file could be closed, use the following command line to see if the output file was produced.

  ```
  .DIRECTORY input-filename.*  RET
  ```

  Use the EDIT/INSPECT command to look at any files displayed by the DIRECTORY command.

9.3 Recovering the Work You Performed and Journaling Any Additional Work

You can recover the work you performed and at the same time produce a new journal file that will contain the work you recovered and any additional work you might perform. A command line to do that has the general format:

EDIT/JOURNAL/RECOVER filespec[/option]

or

EDIT/JOURNAL:jou-filespec/RECOVER:jou-filespec filespec[/option]

If the new journal file and the old journal file (the recovery file) have the same file name and file type, KED changes the file type of the old journal file to JBK. For example, you can type:

```
.EDIT/JOURNAL/RECOVER WEEK.TXT  RET
```

or

```
.EDIT/JOURNAL:RETAIN.REC/RECOVER:RETAIN.REC WEEK.TXT   RET
```

In either example, the old journal file will have the JBK file type (WEEK.JBK or RETAIN.JBK). In the first example, the new journal file will have the JOU file type (WEEK.JOU). In the second example, the new journal file will have the REC file type (RETAIN.REC).

9.4 Changing the Journal Process

When you begin your work session as described in Section 9.1, KED counts and stores the text you type and the changes you make in a journal buffer that can contain a minimum of 512 characters (some systems allow a larger number). When the count reaches a certain number of modifications (the default is 10) or when the journal buffer becomes full, KED moves the contents of that buffer to the journal file. Later, when you end your work session with an EXIT or QUIT command, KED deletes the journal file.

The following sections tell you how to:

- Change the modification count that determines when the contents of the journal buffer will be moved to the journal file.
- Move the contents of the journal buffer to the journal file without waiting for the modification count to be reached.
- Save the journal file when you end your work session with an EXIT or QUIT command.

9.4.1 Changing the Modification Count

To change the modification count that determines when the contents of the journal buffer will be moved to the journal file:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SET FREQUENCY number
   ```

   `number` is the new modification count you want KED to use in determining when the contents of the journal buffer will be moved to the journal file. The highest number allowed is 511. Low numbers can affect system performance, depending on the device you specified for the journal file.
3. Press ENTER.

   KED changes the modification count to the number you specified.

In other words:

GOLD COMMAND `SET FREQUENCY number` ENTER

9.4.2 Moving the Contents of the Journal Buffer to the Journal File

To move the contents of the journal buffer to the journal file without waiting for the modification count to be reached:

1. Press GOLD and COMMAND.
2. Type:

   ```
   WRITE JOURNAL
   ```
3. Press ENTER.

   KED moves the contents of the journal buffer to the journal file you specified when you began your work session.

In other words:

GOLD COMMAND `WRITE JOURNAL` ENTER

9.4.3 Saving the Journal File

If you terminate your work session normally (if you end your work session with an EXIT or QUIT command), KED does not save the journal file.

To save the journal file when you plan to end your work session with an EXIT or QUIT command:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SET JOURNAL SAVE
   ```

3. Press ENTER.

   KED will save the journal file you specified when you began your work session, after you enter an EXIT or QUIT command.

In other words:

GOLD COMMAND `SET JOURNAL SAVE` ENTER

If you do not use the SET JOURNAL SAVE command to save the journal file, you can still save it by using one of the following commands to end your work session.

To save the journal file when you decide to end your work session with an EXIT command:

1. Press GOLD and COMMAND.
2. Type:

   ```
   EXIT SAVE
   ```

3. Press ENTER.

   KED saves the output file and the journal file you specified when you began your work session.

In other words:

GOLD COMMAND `EXIT SAVE` ENTER

To save the journal file when you decide to end your work session with a QUIT command:

1. Press GOLD and COMMAND.
2. Type:

   ```
   QUIT SAVE
   ```

3. Press ENTER.

   KED does not save the output file, but it saves the journal file you specified when you began your work session.

In other words:

GOLD COMMAND `QUIT SAVE` ENTER

9.5 Restrictions

The following operations are invalid.

- Using the /JOURNAL or /RECOVER command option with the /INSPECT file specification option
- Using the @ or LOAD command to execute a journal file
- Copying a journal file with the COPY/ASCII command

In addition, any commands that change the journal process must be placed at the end of an initialization file. Any such commands cause all subsequent commands in the file to be skipped unless the file was executed by an EDIT command with the /JOURNAL option.

9.6 Example

This section summarizes most of the information in this chapter by providing an example of the steps necessary for recovering all the work you performed before a system crash or similar malfunction.

Each step has two parts.

1. An instruction to complete
2. The keys you must type to complete the instruction

The keys you must type are presented in a horizontal line, and the strings of letters and numbers you must type are not enclosed by boxes.

1. Begin your work session to edit file WEEK.TXT.

   ```
   .EDIT/JOURNAL WEEK.TXT  RET
   ```

   KED displays:

   ```
   Monday was cold and rainy.
   On Tuesday, it snowed.
   Wednesday was sunny and hot.
   On Thursday, the lilacs bloomed.
   ```

2. You add a line of text at the bottom of the file.

   ```
   Friday was just like Monday, cold and rainy.  RET
   ```

3. You decide to decrease the modification count, thereby increasing the number of times that KED moves the contents of the journal buffer to the journal file.

   ```
   GOLD  COMMAND  SET FREQUENCY 5  ENTER
   ```

4. You add another line of text at the bottom of the file.

   ```
   On Saturday, it snowed again and the wind was very strong.  RET
   ```

5. Anticipating a system malfunction, you decide to move the contents of the journal buffer to the journal file without waiting for the new modification count to be reached.

```
GOLD  COMMAND  WRITE JOURNAL  ENTER
```

6. The malfunction occurs. Simulate it by typing:

```
GOLD  COMMAND  QUIT SAVE  ENTER
```

7. After resolving the simulated problem, you restart your system and then recover all the work you performed.

```
.EDIT/RECOVER WEEK.TXT  RET
```

KED displays:

```
Monday was cold and rainy.
On Tuesday, it snowed.
Wednesday was sunny and hot.
On Thursday, the lilacs bloomed.
Friday was just like Monday, cold and rainy.
On Saturday, it snowed again and the wind was very strong._
```

8. Finally, you add another line of text at the bottom of the file and then end your work session.

```
On Sunday, we decided to leave New England.  RET
```

```
GOLD  COMMAND  EXIT  ENTER
```

Chapter 10

Special Operations for Programmers

Chapters 2 through 8 describe KED operations you can use to create and edit any kind of text file. This chapter describes some operations you can use for program development.

- Three ways to insert escape characters (Section 10.1)
- For MACRO–11 and MACRO–32 programs, a renumbering operation (Section 10.2) that renumbers your local symbols
- For any program you write in a structured language, a structured tab operation (Section 10.3) that can:

  Create multiple levels of indentation
  Insert statements in a structured program
  Change the indentation level of a group of statements
  Rewrite indentations to use the most efficient combination of tabs and spaces

Although the KED features described in this chapter are robust in design, some are specialized in nature and, therefore, performed infrequently. For that reason, you should consider closing and reopening a file and thereby making a backup copy, before performing special operations on the file.

10.1 Inserting Escape Characters

KED gives you three ways to insert the escape character.

1. On all terminals, press CTRL/[(CTRL and left square bracket) twice.

   KED displays the pi (π) symbol.

2. On many terminals, press ESC twice.

   KED displays the pi (π) symbol.

3. On all terminals, use the SPECINS function.

   a. Press GOLD.

   b. Type the ASCII decimal equivalent of the escape character (27).

   c. Press GOLD and SPECINS.

      KED displays the pi (π) symbol.

In other words:

CTRL/[CTRL/[
or

ESC ESC
or
GOLD 27 GOLD SPECINS

You can use the following escape characters for the keypad functions listed. Two ways to insert the escape equivalents in a file are:

- You can insert an escape character as described above and then type the indicated characters.
- You can issue a single escape character by pressing CTRL/[or ESC once, press CTRL and, while pressing it, press the function key you want to insert. For example, to insert the ADVANCE function:

  CTRL/[CTRL/*x* (where *x* is the ADVANCE function key)

  All functions on the keypad can be inserted in that manner.

Table 10–1: Keypad Function Escape Equivalents

Keypad Function	Escape Equivalent	Operation
ADVANCE	πOt	Changes the direction of cursor movement to the right and downward. Enters a search model, specifies a search direction, and starts a search.
APPEND	πOy	Deletes selected text and stores it at the end of the paste buffer.
BACKUP	πOu	Changes the direction of cursor movement to the left and upward. Enters a search model, specifies a search direction, and starts a search.
BLINE	πOp	Moves the cursor to the beginning of a line.
BOTTOM	πOt	Moves the cursor to the end-of-file marker.
CHAR	πOs	Moves the cursor to the next or preceding character.
CHNGCASE	πOPπOq	In selected text, changes all lowercase letters to uppercase and all uppercase letters to lowercase. Changes the case of letters in a search target. In unselected text, changes the case of the cursor's character.
COMMAND	πOPπOw	Displays the `Command:` prompt, which allows specification of a command from the keyboard.

Table 10–1 (Cont.): Keypad Function Escape Equivalents

Keypad Function	Escape Equivalent	Operation
CUT	πOv	Deletes selected text and stores it in the paste buffer.
DELCHAR	πOl	Deletes the cursor's current character and stores it in the character buffer.
DELEOL	πOPπOr	Deletes to the next line terminator and stores the string in the line buffer.
DELLINE	πOS	Deletes through the next line terminator and stores the string in the line buffer.
DELWORD	πOm	Deletes to the first character of the next word and stores the string in the word buffer.
ENTER	πOM	Executes a command and erases the command and the command prompt.
		Enters a search model without starting a search. FINDNEXT will start the search.
		Restores the display after the HELP function has been used.
EOL	πOr	Moves the cursor to a line terminator.
FILL	πOPπOr	Reformats the lines in selected text.
FIND	πOPπOR	Displays the `Model:` prompt, which allows the specification of a search string.
FINDNEXT	πOR	Uses the keypad directional mode to search for another occurrence of a target that matches a search model.
		Starts a search after the ENTER function has been used.
GOLD	πOP	Specifies the lower function of the two functions on a keypad key.
		Allows the use of the GOLD keyboard-key functions.
HELP	πOQ	If an error occurred, displays an error message.
		If no error occurred or if KED is displaying an error message, displays a keypad diagram.
		If the keypad diagram is displayed, displays summaries of KED commands and functions.
OPENLINE	πOPπOp	Breaks the current line by inserting a new-line terminator at the cursor's right.

Table 10–1 (Cont.): Keypad Function Escape Equivalents

Keypad Function	Escape Equivalent	Operation
PAGE	πOw	Moves the cursor to the top of a page. The page default is a form-feed character.
PASTE	πOPπOv	Inserts text from the paste buffer.
REPLACE	πOPπOy	Deletes and discards selected text. Deletes and discards a search target and inserts a character string from the paste buffer.
RESET	πOPπOn	Cancels a GOLD function or a text selection.
SECTION	πOx	Moves the cursor to the top of a section. The section default is 16 lines.
SELECT	πOn	Marks one end of selected text.
SPECINS	πOPπOs	Inserts a nonprinting character.
SUBSTITUTE	πOPπOM	When the cursor is at a search target, completes the REPLACE function and then the FINDNEXT function.
TOP	πOPπOu	Moves the cursor to the top of a file.
UNDELCHAR	πOPπOl	Inserts a character from the character buffer.
UNDELLINE	πOPπOS	Inserts a character string from the line buffer.
UNDELWORD	πOPπOm	Inserts a character string from the word buffer.
WORD	πOq	Moves the cursor to the beginning of a word.

10.2 Renumbering MACRO–11 and MACRO–32 Local Symbols

Good programming practice requires that local symbols within each local symbol block (LSB) begin with the symbol 10$ and continue in increments of ten—20$, 30$, 40$, and so on. In the course of program development, that order can be upset and the symbols have to be renumbered.

You can use the LOCAL command to renumber your MACRO–11 or MACRO–32 local symbols, but you must observe some rules and restrictions.

10.2.1 LOCAL Command

The LOCAL command lets you renumber the local symbols in your MACRO–11 source program.

1. Press GOLD and COMMAND.

2. Type:

   ```
   LOCAL [start] [increment]
   ```

   `start` is the optional starting value. You can type any whole number from 1 to 32767. The default is 10.

   `increment` is the optional incrementing value. You can type any whole number from 1 to 32767. The default is 10.

3. Press ENTER.

   KED renumbers your local symbols. KED also renumbers all references to each local symbol, including references within comments.

In other words:

```
GOLD COMMAND LOCAL [start] [increment] ENTER
```

For example, to renumber all the local symbols in your source program, beginning with 100$ and incrementing each subsequent symbol by 100:

```
GOLD COMMAND LOCAL 100 100 ENTER
```

To renumber several LSBs that use the same local symbols:

1. Renumber each LSB separately, using a starting local symbol value that will not overlap any values in the other LSBs.
2. Combine the LSBs in your program.
3. Renumber the combined LSB.

For example, to renumber the local symbols within two LSBs, use a procedure such as the following.

1. Renumber one of the LSBs with the LOCAL 1000 command.
2. Combine the LSBs by using the CUT and PASTE operations.
3. Renumber the combined LSB with the LOCAL 10 command.

10.2.2 Rules and Restrictions

When renumbering local symbols, observe the following rules and restrictions.

- Do not abort the renumbering operation by pressing CTRL/C. Local symbol renumbering alignment is destroyed, making the file unusable.
- Large local symbol blocks can take some time to renumber.
- Begin renumbering an LSB at the program line below the directive that enables the LSB you are renumbering. Place the cursor on that line and issue the LOCAL command.
- KED recognizes (detects and changes) only local symbols that have the form:

  ```
  [white_space]number$:
  ```

  `white_space` is a line terminator and one or more optional spaces or tabs.

`number` is the local symbol number. It cannot be any longer than six digits.

For example, this local symbol is formatted correctly.

```
30$:
```

- KED recognizes only references to local symbols that have the form:

  ```
  character number$
  ```

  `character` is any non-Radix–50 character.

  `number` is the local symbol number.

  For example, this local symbol reference is formatted correctly.

  ```
  BCS    30$
  ```

- KED recognizes only the following LSBs:
  - LSBs that are explicitly included in source code. LSBs that are part of macro expansions are not recognized.
  - LSBs that begin with an .ENABL LSB, .ENABLE LSB, or .ENABLE LOCAL_BLOCK directive and end with a .DSABL LSB, .DISABLE LSB, or .DISABLE LOCAL_BLOCK directive. Renumbering is also terminated in an LSB when KED finds another directive to enable an LSB before finding a directive to disable the LSB; the second enabling directive is treated as a disabling directive.
  - LSBs that begin with .ENABLE LOCAL_SYMBOL_BLOCK and with .DISABLE LOCAL_SYMBOL_BLOCK with MACRO–32.
  - LSBs that begin with a nonlocal symbol definition and end with another nonlocal symbol definition, an .ENABL LSB directive, or a .DSABL LSB directive.

  MACRO–11 accepts other LSBs, but KED cannot recognize them. For example, KED cannot recognize the .PSECT directive as the end of an LSB, even though it is a valid LSB terminator for MACRO–11 programs. Therefore, KED continues to renumber the local symbols below a .PSECT directive until it finds an LSB terminator it recognizes.

10.3 Using the Structured Tab Operation

KED's structured tab operation lets you:

- Create multiple levels of indentation (Section 10.3.1)
- Insert statements in a structured program (Section 10.3.2)
- Change the indentation level of a group of statements (Section 10.3.3)
- Rewrite indentations to use the most efficient combination of tabs and spaces (Section 10.3.4)

Those features are enabled by the SET TABS command described in Section 10.3.1.1.

10.3.1 Creating Multiple Levels of Indentation

To create a program with multiple levels of indentation:

1. Specify an indentation value (the number of columns separating each level of indentation) and initialize the level counter with a value of 1 to indicate the first level of indentation. That operation also enables all the structured tab features.
2. Increment or decrement the level counter to specify the level of indentation at which you want to type in code. That operation moves the indentation right or left.

10.3.1.1 Specifying an Indentation Value and Initializing the Level Counter (Enabling Structured Tab Features)

To specify an indentation value and initialize the level counter at 1:

1. Press GOLD and COMMAND.
2. Type:

   ```
   SET TABS [indent]
   ```

   `indent` is any whole number from 1 to 32767 (8 or a multiple of 8 will ensure the correct alignment of statement comments if they are preceded by horizontal tabs). The default is 4.
3. Press ENTER.

   KED stores the indentation value and sets the level counter to 1.

In other words:

```
GOLD COMMAND SET TABS [indent] ENTER
```

Once you enter the SET TABS command, you can type in code at the first level of indentation.

1. Move the cursor to the beginning of a new line.
2. Press TAB.

   KED advances the cursor to the first level of indentation.
3. Type in your code.

For example, to specify an indentation of eight columns and initialize the level counter:

GOLD COMMAND `SET TABS 8` ENTER
Move the cursor to a new line.
TAB
KED advances the cursor eight columns to the first level of indentation.
Type in code at that level.

If you do not move the cursor to the beginning of a new line (if a space or horizontal tab is to the left of the cursor) before you press TAB, KED responds as if the structured tab features were disabled.

To cancel the indentation value and the initialization of the level counter:

```
GOLD COMMAND SET NOTABS ENTER
```

That command disables all structured tab features, including the TABS ADJUST commands (Sections 10.3.3 and 10.3.4).

10.3.1.2 Incrementing and Decrementing the Level Counter (Moving the Indentation Right and Left)

Incrementing the level counter moves your indentation to the right. Decrementing the level counter moves your indentation to the left.

Incrementing Level Counter—Moving Indentation Right

To move the cursor to higher levels of indentation (increment the level counter):

1. Issue the SET TABS command as described in Section 10.3.1.1.
2. Press GOLD.
3. Type:

   ```
   E
   ```

   KED increments the level counter by 1.
4. To continue incrementing the level counter, repeat steps 3 and 4.
5. Press TAB.

   KED moves the cursor to the level you specified.
6. Type a line of code and press RETURN.

   KED moves the cursor to the beginning of the next line.
7. To begin another line of code at the same indentation level, press TAB.

In other words:

```
GOLD COMMAND SET TABS [indent] ENTER
GOLD (E)
TAB line-of-code RET
```

If you do not move the cursor to the beginning of a new line (if a space or horizontal tab is to the left of the cursor) before you press TAB, KED responds as if the structured tab features were disabled.

Decrementing Level Counter—Moving Indentation Left

To move the cursor to lower levels of indentation (decrement the level counter):

1. Issue the SET TABS command as described in Section 10.3.1.1.
2. Press GOLD.
3. Type:

   ```
   D
   ```

KED decrements the level counter by 1.

4. To continue decrementing the level counter, repeat steps 3 and 4.

5. Press TAB.

   KED moves the cursor to the level you specified.

6. Type a line of code and press RETURN.

   KED moves the cursor to the beginning of the next line.

7. To begin another line of code at the same indentation level, press TAB.

In other words:

```
GOLD COMMAND SET TABS [indent] ENTER
GOLD D
TAB line-of-code RET
```

If you do not move the cursor to the beginning of a new line (if a space or horizontal tab is to the left of the cursor) before you press TAB, KED responds as if the structured tab features were disabled.

10.3.2 Inserting Statements in a Structured Program

During the development of a program, you may want to insert additional statements in an indented section of existing statements.

1. Issue the SET TABS command as described in Section 10.3.1.1.

2. Move the cursor to the indentation level of the section in which you want to insert additional statements.

3. Press GOLD.

4. Type:

   ```
   A
   ```

   KED sets the indentation level.

5. Move the cursor to the left margin.

6. Press TAB.

   KED moves the cursor to the indentation level you set with the A key.

   KED calculates the level of indentation by dividing the indentation value (which you set with the SET TABS command) into the number of the column where the cursor is located. The indentation value must divide evenly into the column number. If it does not, KED signals that an error occurred and ignores its calculation. If you press HELP, KED displays:

   ```
   TAB indent value wrong for align.
   ```

In other words:

```
GOLD COMMAND SET TABS [indent] ENTER
```

Move the cursor to the indentation level.
GOLD A
Move the cursor to the left margin.
TAB

For example, to add additional statements to a group of statements at the fourth level of indentation when the indentation value is 5:

Move the cursor to the column in which the statements begin.
GOLD A
Since the cursor is at column 20, KED calculates the indentation level as 4.
Move the cursor to the left margin.
TAB
KED moves the cursor to the fourth level of indentation.

10.3.3 Changing the Indentation Level of a Group of Statements

In developing a program you may want to change the indentation level of a group of statements.

1. Issue the SET TABS command as described in Section 10.3.1.1.
2. Move the cursor to the left margin of the line containing the first statement in the group.
3. Press SELECT.
4. Move the cursor to the character following the last statement in the group.
5. Press GOLD and COMMAND.
6. Type:

   ```
   [TABS] ADJUST [level]
   ```

   `level` is the number by which you want to increment or decrement the level counter. To increment the level counter (obtain a higher level of indentation), type any number from +1 to +50. To decrement the level counter (obtain a lower level of indentation), type any number from –1 to –50. The default is 0 (see Section 10.3.4).
7. Press ENTER.

   KED reformats all the selected lines to begin at the level you specified.

In other words:

GOLD COMMAND `SET TABS [indent]` ENTER
Select the group of statements.
GOLD COMMAND `ADJUST [level]` ENTER

When you adjust the indentation of selected lines, KED cannot maintain the position of comments that are to the right of executable statements in the selection, unless the indentation value you used in enabling structured tabs (Section 10.3.1.1) is 8 or a multiple of 8 and the comments are preceded by tab characters.

To cancel the SET TABS command:

GOLD COMMAND `SET NOTABS` ENTER

10.3.4 Rewriting Indentations to Use the Most Efficient Combination of Tabs and Spaces

KED can replace the existing combination of tabs and spaces at the beginning of text lines with the most efficient combination possible, perhaps reducing the amount of media required to store the file. The indentation itself is not changed. This feature is useful only with text lines not written with the aid of structured tabs and should not be used with text lines that use structured tabs.

When using structured tabs to indent a line, KED:

Inserts a tab character for every eight columns in the indentation
Fills in any remaining columns with single spaces

To optimize the indentation of lines that you typed in without the aid of structured tabs:

1. Issue the SET TABS command as described in Section 10.3.1.1.
2. Move the cursor to the left margin of the first line you want to optimize.
3. Press SELECT.
4. Move the cursor to the left margin of the line following the last line you want to optimize.
5. Press GOLD and COMMAND.
6. Type:

   ```
   [TABS] ADJUST [0]
   ```
7. Press ENTER.

   KED does not change the amount of indentation of the lines you selected, but replaces the spaces and tabs at the beginning of each line with the most efficient combination of spaces and tabs possible for that amount of indentation.

In other words:

GOLD COMMAND `SET TABS [indent]` ENTER
Select the lines to be optimized.
GOLD COMMAND `ADJUST` ENTER

When you adjust the indentation of selected lines, KED cannot maintain the position of comments that are to the right of executable statements in the selection, unless the indentation value you used in enabling structured tabs (Section 10.3.1.1) is 8 or a multiple of 8 and the comments are preceded by tab characters.

To cancel the SET TABS command:

GOLD COMMAND `SET NOTABS` ENTER

10.3.5 Examples of the Structured Tab Operation

This section provides examples of the structured tab operation in a short section of a FORTRAN program. By creating a new file and then following the instructions for each step, you can see the processes that take place.

1. Enable structured tabs and set the basic indentation to 4.

   ```
   GOLD  COMMAND  SET TABS 4  ENTER
   ```

2. With the cursor at column 1, type:

   ```
   TAB  DO 100 NUM = START, FINISH  RET
   ```

   KED indents the statement to column 5 with 4 spaces, because the basic indentation is 4 and the level counter is 1. The indentation is created with spaces, because it is not large enough to permit any tab characters to be used.

3. Increase the level counter to 2.

   ```
   GOLD  E
   ```

4. With the cursor at column 1, type:

   ```
   TAB  SUM = 0  RET
   TAB  SUM = SUM + ARRAY (NUM)  RET
   100  TAB  CONTINUE  RET
   ```

   The programming error with the SUM = 0 statement will be corrected in a later step.

   KED indents the first two statements to column 9 with a tab character (equivalent to 8 spaces), because the level counter is now 2. Each indentation is created with a tab character, because KED calculates that the tab character will have the same effect as 8 spaces.

5. Decrease the indentation.

   ```
   GOLD  D
   ```

6. With the cursor at column 1, type:

   ```
   TAB  TYPE *, SUM  RET
   TAB  END  RET
   ```

   KED indents the statements to column 5 by inserting 4 spaces, because the level counter is now 1.

   Your statements should now look like this.

   ```
       DO 100 NUM = START, FINISH
           SUM = 0
           SUM = SUM + ARRAY (NUM)
   100     CONTINUE
       TYPE *, SUM
       END
   ```

7. With the cursor at column 1 of the DO statement, indent the six statements 4 additional spaces.

SELECT GOLD BOTTOM GOLD COMMAND TABS ADJUST +1 ENTER

KED increases each indentation by 4 columns, by calculating and inserting the most efficient combination of tabs and spaces for each statement.

8. The SUM = 0 statement is inside the DO loop. Fix that error by moving the SUM = 0 statement above the DO statement.

   Your statements should now look like this.

```
            SUM = 0
        DO 100 NUM = START, FINISH
            SUM = SUM + ARRAY (NUM)
100         CONTINUE
        TYPE *, SUM
        END
```

9. With the cursor at column 1 of the SUM = 0 statement, decrease the indentation of the SUM = 0 statement.

   SELECT ADVANCE BLINE GOLD COMMAND TABS ADJUST -1 ENTER

10. With the cursor on C of the word CONTINUE, insert a new statement in the DO loop and align it with the other statements in that loop.

    a. Set the level counter properly.

       GOLD A

    b. Move the cursor to the beginning of the 100 CONTINUE statement and create an open line for the new statement.

       BACKUP BLINE GOLD OPENLINE

       Your statements should now look like this.

```
        SUM = 0
        DO 100 NUM = START, FINISH
            SUM = SUM + ARRAY (NUM)

100         CONTINUE
        TYPE *, SUM
        END
```

    c. Type the new statement.

       Tab IF (ARRAY(NUM) .LT. 0) TYPE *,'Low value at ',NUM

       KED indents the new statement to column 13. The total indentation is the equivalent of 12 spaces (3 times 4), because the GOLD A function set the level counter to 3 and the basic indentation is 4.

       Your statements should now look like this.

```
        SUM = 0
        DO 100 NUM = START, FINISH
            SUM = SUM + ARRAY (NUM)
            IF (ARRAY(NUM) .LT. 0) TYPE *,'Low value at ',NUM
100         CONTINUE
        TYPE *, SUM
        END
```

Appendix A
RT–11 Operating System Procedures

This appendix describes procedures that are specific to the RT–11 operating system and features that are only available with RT–11. You should also read at least the first three chapters of the *Introduction to RT–11* to learn about RT–11 before using KED.

RT–11 distributes two versions of the keypad editor, KED and KEX. KED and KEX are essentially the same and all references in this manual to KED apply also to KEX unless indicated otherwise.

KED is the default editor for the SJ and FB monitors, and is run under those monitors when you issue the EDIT command. Also, you can run KED directly as the background or the foreground job, as described in this appendix.

KEX is the default editor for the XM monitor and can run only under that monitor. KEX is a virtual KED program. Its functions and commands are the same and provide the same operations as the KED functions and commands. Under the XM monitor, KEX is run when you issue the EDIT command. Also, you can run KEX directly as the background, the foreground, or as a system job, as described in this appendix.

NOTE

RT–11 supports KED as described in this manual's chapters. All commands, features, and functions are supported as described.

A.1 Installing KED

The PDP–11 keypad editor is installed on your system when you install RT–11.

KED requires the file KED.SAV, which is installed on your system device as part of the RT–11 automatic installation procedure.

If you are running the XM monitor, RT–11 automatically uses KEX, the virtual version of KED. KEX requires the file KEX.SAV, which is also installed on your system device as part of the RT–11 automatic installation procedure.

If you installed RT–11 manually, be sure the appropriate version of KED (KED or KEX) resides on your system (SY) device.

A.1.1 System Requirements

KED is supported on only VT100 series compatible terminals (VT100 series, VT200 series, and VT300 series). You cannot use KED with a hardcopy terminal or a terminal that is not compatible with the VT100 series, but you can use EDIT.SAV, the nonkeypad (line) editor described in the *RT–11 System User's Guide*.

KED works as documented under all distributed RT–11 monitors.

If you have performed a system generation procedure (SYSGEN) and built support for a multiterminal monitor and you are running KED on a terminal other than 0, issue the following commands or include them in your startup command file:

```
.SET TT CONSOL=n RET
.SET TT NOCRLF RET
```

where *n* is the logical unit number of your terminal.

A.2 Running KED

You can run KED in three ways:

1. By using the DCL command EDIT, as described in the chapters of this manual
2. By using the CCL commands KED or KEX

   Although supported, CCL command syntax is not particularly useful when running KED, and is not specifically described in this manual. DCL command syntax is recommended instead of CCL.
3. By using the CSI commands R, RUN, FRUN, or SRUN, and specifying the KED or KEX utility

   CSI command syntax is only required if you are running KED or KEX as the foreground (FRUN) or as a system (SRUN) job. Running KEX as a system job is described, with examples, in the *Introduction to RT–11*. CSI command syntax for KED is described in this appendix. Using CSI command syntax in general is described in the *RT–11 System Utilities Manual*.

The principal differences between running KED from the DCL, CCL, or CSI command syntax are:

- The simplicity (or complexity) of the command syntax. In general, DCL is the simplest command syntax, and is therefore recommended. DCL syntax (the EDIT command) is used throughout this manual, except Section A.2.2 in this appendix. The only time you cannot use DCL syntax is when you run KED/KEX as the foreground or as a system job; then, you must use CSI syntax as described in Section A.2.2.
- How options are specified. Options for the KED DCL command syntax are as described in the chapters. Options for CCL and CSI command syntax are described in Section A.2.2.
- How you return to the RT–11 prompt and whether special settings you made are preserved when you end your work session.

If you use the KED DCL or CCL command syntax, KED returns you to the RT–11 prompt when you end your work session. Any special settings you made and anything you stored in KED's buffers are lost.

If you use the KED CSI command syntax, KED returns you to the CSI level and displays the KED utility prompt (*). Unless you use a journal file as described in Section A.2.2.3, any special settings you made and anything you stored in KED's buffers are preserved across work sessions until you abort the utility.

A.2.1 Command Default Values

Item	Default
Text editor	KED for the SJ and FB monitors or KEX for the XM monitor.
KED volume	For the EDIT, KED, KEX, R, FRUN, and SRUN commands, the system volume, SY. For the RUN command, the default storage volume, DK.
Editing process	Unless you specify a different volume, KED backs up the file you specify on the same volume that contains your input file.
Input volume	The default storage volume, DK
Input file name	The default is the output file name.
Input file type	The default is the output file type.
Output volume	The default storage volume, DK
Output file name	The default is the input file name, if the input file name is specified.
Output file type	MAC when editing a file or LST when inspecting a file. A customization patch in the *RT–11 Installation Guide* lets you change those defaults.
Maximum output file size	Depends on the number of contiguous free blocks on your output volume. KED allocates the larger of the following sizes. One-half of the largest number of contiguous free blocks All of the second largest number of contiguous free blocks

A.2.2 Using the R, RUN, FRUN, and SRUN Commands

You can begin any work session with an R, RUN, FRUN, or SRUN command. Observe the following rules:

- If KED.SAV or KEX.SAV is on your default storage volume (DK), you do not have to specify a device for the RUN command. For example, with KED.SAV on DK, the following commands are equivalent:

```
.RUN KED RET
.RUN DK:KED RET
```

If KED.SAV or KEX.SAV is on your system volume, you do not have to specify a device for the R, FRUN, or SRUN command. For example, with KED.SAV on SY, the following commands are equivalent.

```
.R KED  RET
.R SY:KED  RET
```

If KED.SAV or KEX.SAV is on another volume, you have to include the device specification for that volume. The following examples show two device specifications.

```
.RUN DL0:KED  RET
.RUN DU0:KED  RET
```

- You must specify the file type (.SAV) when you run KED or KEX with the FRUN or SRUN command, as follows:

```
.FRUN KED.SAV  RET
.SRUN KEX.SAV  RET
.SRUN DU0:KEX.SAV  RET
```

A.2.2.1 The CSI File Specification String

After you enter an R, RUN, FRUN, or SRUN command, KED displays its utility prompt (*) and waits for a CSI file specification string. For example:

```
.FRUN KED.SAV  RET
*
```

See the *RT–11 System Utilities Manual* for a description of the command string interpreter.

DK is the default device for all CSI file specifications. The *Introduction to RT–11* and *RT–11 System User's Guide* include details on permanent device names and ways you can assign the logical name DK to different physical devices.

A.2.2.2 Beginning Your Work Session

A complete CSI file specification string for beginning a work session has the general format:

output-filespec[/option]=input-filespec[/option]

output-filespec — should be a standard file specification in the form *[dev:]filnam.typ*. You must use it when:

- Creating new files (no existing input file)
- Editing files and using new output file names

input-filespec

should also be a standard file specification in the form *[dev:]filnam.typ*. You must use it when:

1. Editing files with automatic backup
2. Inspecting existing files
3. Editing files and using new output file names

/option For *output-filespec*, */option* can be /A:size or /C.

For *input-filespec*, */option* can be /A:size, /I, /N, or /Y.

- Use /A:size to specify the size of your output file, as described in Section A.2.2.2.1 and Section A.2.2.2.4
- Use /C to create a new file, as described in Section A.2.2.2.2
- Use /I to inspect an existing file, as described in Section A.2.2.2.3
- Use /N to prevent initialization file execution, as described in Section A.2.2.4
- Use /Y to suppress KED queries, as described in Section A.2.2.4

A.2.2.2.1 Editing Files with Automatic Backup

The CSI file specification string for editing files with automatic backup can be of two forms:

output-filespec[/A:size]=input-filespec

or

input-filespec[/A:size]

where *size* is the number of blocks to be allocated for the file. The maximum is 32767.

In the first form, the specifications for the output file and input file must be the same if you want automatic backup. For example, at the end of an editing session that you begin with the following commands, KED stores the previous version of EXAMP.FOR under the name EXAMP.BAK and the new version of the file under the name EXAMP.FOR.

```
.FRUN KED.SAV RET
*DU1:EXAMP.FOR=DU1:EXAMP.FOR RET
```

Typing only the specification for the input file that you want to edit is the shortest way to specify automatic backup. The following example has the same effect as the preceding example.

```
.FRUN KED.SAV RET
*DU1:EXAMP.FOR RET
```

A.2.2.2.2 Creating New Files

The CSI file specification string for creating a new file has the form:

output-filespec/C

If another file with the same specification exists, KED responds with a warning message. At that time, you can cancel your work session or you can continue the session, deleting the existing file, and creating a new file with the old name.

The following example creates a new file, EXAMP.BAS, on default device DK.

```
.FRUN KED.SAV RET
*EXAMP.BAS/C RET
```

KED shows an empty screen with the cursor at the upper left corner.

The next example creates a new file, ARLO.FOR, on diskette drive DX0. KED warns that another file with the same name exists. It is preserved because you respond N (for no) to the warning.

```
.FRUN KED.SAV RET
*DX0:ARLO.FOR/C RET
?KED-W-Output file exists - Continue (Y,N)? N RET
*
```

If you had responded Y (for yes), the existing version of ARLO.FOR would have been deleted and KED would have shown an empty screen with the cursor at the upper left corner.

A.2.2.2.3 Inspecting Existing Files

The CSI file specification string for inspecting an existing file has the form:

input-filespec/I

The following example begins the inspection of a file, RPT296.TXT, on default device DK.

```
.FRUN KED.SAV RET
*RPT296.TXT/I RET
```

KED displays the top 24 screen lines of DK:RPT296.TXT.

A.2.2.2.4 Editing Files and Using New Output File Names

The CSI file specification string for editing files and using new output file names has the form:

output-filespec[/A:size]=input-filespec

where *size* is the number of blocks to be allocated for the output file. The maximum is 32767.

The specifications for the output file and input file must be different if you want KED to store the edited version of the file under a new name.

For example, at the end of a work session that you begin with the following commands, KED stores the edited version of TEMP.DAT under the name EXPT44.DAT.

```
.FRUN KED.SAV RET
*EXPT44.DAT=TEMP.DAT RET
```

A.2.2.3 Using Journal Files to Protect Your Work Session

You can create a journal file for your work session by using a second output file specification in your CSI file specification string. Then, if you experience a power-down or other system malfunction, you can recover the work you performed by using a second input file specification in your CSI file specification string. You can also journal any additional work you might perform by using a second output file specification, as well as a second input file specification, in your CSI file specification string.

The recovery operation begins and ends before an initialization file (Section A.2.2.4), or your new editing commands can be processed by KED.

A.2.2.3.1 The CSI File Specification String for Journal Files

A complete CSI file specification string for using journal files has the general format:

output-filespec,journal-filespec=input-filespec,journal-filespec[/option]

output-filespec — is the output file specification described in Sections A.2.1 and A.2.2.2.

input-filespec — is the input file specification described in Sections A.2.1 and A.2.2.2.

journal-filespec — On the output side, *journal-filespec* is the specification of a journal file that will contain commands for a subsequent recovery operation. The default device is DK. The default file name is the name of your input file. The default file type is JOU.

On the input side, *journal-filespec* is the specification of another journal file that is used to recover the work you performed in a previous work session. The default device is DK. You must specify the file name. The default file type is JOU.

/option — can be one or more of the file specification options listed in Section A.2.2.2.

A.2.2.3.2 Beginning Your Work Session

The CSI file specification string for creating a journal file has the form:

output-filespec,journal-filespec=input-filespec

output-filespec — is the output file specification described in Sections A.2.1 and A.2.2.2. If not specified, its place must be marked by including the comma.

journal-filespec — Must be specified.

input-filespec — is required.

The following example invokes KED to edit MYFILE.RNO and specifies SAVEIT.JOU as a journal file to reside on DL1.

```
.FRUN KED.SAV RET
*,DL1:SAVEIT=MYFILE.RNO RET
```

Whenever you begin a work session that creates a journal file, any special settings you made and anything you stored in KED's buffers are lost after you end that work session, even if you did not press CTRL/C.

See the *RT–11 Installation Guide* for a customization patch that always creates a journal file whenever you begin a work session.

A.2.2.3.3 Recovering the Work You Performed and Journaling Any Additional Work

Once you create a journal file, you can use it to recover the work you performed in a previous work session. And, at the same time, you can create a second journal file to preserve your additional work (and that of the recovery operation) in your current work session.

The recovery operation starts at the beginning of your work session, before KED can execute an initialization file (Section A.2.2.4) and before you can perform any additional work.

The CSI file specification string for using a journal file to recover the work you performed and for creating a second journal file has the form:

output-filespec,journal-filespec=input-filespec,journal-filespec

output-filespec — is optional. If you use it, you must type the equal sign.

input-filespec — is required.

journal-filespec — On the output side, *journal-filespec* is the specification of a new journal file that will be created for your work session. Its use is optional. If you use it, you must type the comma and the equal sign.

On the input side, *journal-filespec* is the specification of the existing journal file that is used to recover the work you performed in a previous work session.

The following example invokes KED to edit MYFILE.RNO and specifies that SAVEIT.JOU on DL1 will be executed first in the work session. The result, when the work session is ended, will be a new MYFILE.RNO containing all the work performed, a backup file named MYFILE.BAK, and the original SAVEIT.JOU on DL1.

```
.FRUN KED.SAV RET
*MYFILE.RNO,DL1:SAVEIT.JOU RET
```

The next example invokes KED to edit MYFILE.RNO and execute MYFILE.JOU on DK. After the work session is ended, a new MYFILE.JOU will be on DK.

```
.FRUN KED.SAV RET
*,MYFILE=MYFILE.RNO,MYFILE RET
```

A.2.2.4 Using Initialization Files

After performing a recovery operation (if any was specified) and before letting you begin your work, KED looks for initialization file DK:KEDINI.KED. If that file does not exist, KED looks for SY:KEDINI.KED. If one of those files exists, KED executes it. If those files do not exist, KED does not signal an error.

You can prevent initialization file execution by using the /N option in your CSI file specification string:

output-filespec=input-filespec/N

For example:

```
.FRUN KED.SAV RET
*MYFILE.RNO,MYFILE/N RET
```

You can also specify another initialization file (not KEDINI.KED) by using a third input file specification in your CSI file specification string. The second comma on the input side marks the position for a journal file that is not being specified. If the comma is not included, KED incorrectly assumes the initialization file is a journal file.

output-filespec=input-filespec,,init-filespec[/option]

output-filespec — is the output file specification described in Sections A.2.1 and A.2.2.2.

input-filespec — is the input file specification described in Sections A.2.1 and A.2.2.2.

init-filespec — is the specification for your initialization file. You must type the two commas and a file name. The default device is DK. The default file type is KED.

/option — can be one or more of the file specification options listed in Section A.2.2.2.

For example, the following command line calls the initialization file, MYINIT.KED, before opening MYFILE.RNO for editing:

```
.FRUN KED.SAV RET
*MYFILE.RNO,,MYINIT.KED RET
```

If your initialization file does not exist, KED displays:

```
?KED-W-Command file not found - Continue (Y,N)?
```

Type Y and press RETURN to begin your work session without executing any initialization file. Type N and press RETURN to return to the KED utility prompt.

If you want to suppress that query and any others KED could generate (for example, `?KED-W-Output file exists - Continue (Y,N)?`), use the /Y option in your CSI file specification string:

output-filespec=input-filespec,,init-filespec/Y

or

output-filespec=input-filespec/Y

For example:

```
.FRUN KED.SAV  RET
*MYFILE.RNO/Y  RET
```

A.2.3 Specific RT–11 KED Information

This section describes information for running KED that is specific to the RT–11 operating system.

A.2.3.1 Wildcard Characters

You cannot use wildcard characters (* or %) in file specifications.

A.2.3.2 Running KED from Indirect Files

When you set the ERROR condition to WARNING (SET ERROR WARNING) and then run KED from an indirect command file, ending the work session with the QUIT command causes RT–11 to abort the indirect command file.

A.2.3.3 Overwriting Existing Files

Except when editing with automatic backup, KED displays the following message if you specify an output file that already exists on your output volume:

```
?KED-W-Output file exists - Continue (Y,N)?
```

When that message appears, you can end your work session and preserve the existing file by responding with N (for no). Responding to that message with Y (for yes) continues the work session. Subsequently ending the work session by issuing the EXIT command deletes the existing file and creates a new file with the old name. Subsequently ending the work session by issuing the QUIT command preserves the existing file, but loses any edits made during that work session.

In the following example, KED warns that diskette DX1 contains a file named MAYQUE.BAS. Assume you respond to the warning prompt with Y:

```
.EDIT/OUTPUT:DX1:MAYQUE.BAS DX0:PG451.BAS  RET
?KED-W-Output file exists - Continue (Y,N)? Y  RET
```

If you then end the work session with the EXIT command, MAYQUE.BAS is deleted. But, if you stop KED with the QUIT command, MAYQUE.BAS is not deleted.

A.2.3.4 Output File Size Messages

KED searches your output volume for contiguous free space in which to store the file you are creating. Then, KED allocates from that space for your work session.

In the following cases, KED displays messages about the space it has allocated. (Appendix F tells you what to do when the messages appear.)

- When your output volume does not contain any contiguous free space that is as large as your input file, KED displays the following message and returns to the RT–11 prompt or the KED utility prompt.

  ```
  ?KED-F-Output file shorter than input file
  ```

- When the largest amount of contiguous free space on your output volume is less than 10 blocks larger than your input file, KED displays the following message.

  ```
  ?KED-W-Only MMMMMM blocks available for insertions - Continue (Y,N)?
  ```

  KED also displays that message when you create a new file and your output volume contains less than 10 blocks of contiguous free space.

- When you are adding text to a file and the file becomes as large as the space allocated for it, KED refuses to accept more text and signals you by sounding a beeper or reversing the text and background on your terminal. If you then use the HELP function, KED displays:

  ```
  Insert finds file full
  ```

A.2.3.5 Using the Wrong Kind of Terminals

KED supports only video terminals that are compatible with the VT100 series. You can use the following recovery procedures if you attempt to run KED on a noncompatible terminal.

A.2.3.5.1 Hardcopy Terminals

If you accidentally run KED from a hardcopy terminal, use the following procedure to recover control of your system.

1. Press CTRL/C several times if the EDIT command prompt (`File?`) or the CSI prompt (`*`) was displayed.
2. Reboot your system if you:

   Used the EDIT command and included a complete file specification
   Responded to the CSI prompt with a file specification

Continue working normally when the RT–11 prompt appears.

A.2.3.5.2 VT50 Series Video Terminals

If you accidentally run KED from a VT52 or VT55 terminal, repeat the following procedure until you recover control of your system.

1. Press CTRL/C several times.
2. Turn your terminal off, then on.
3. Press RETURN.

Continue working normally when the RT–11 prompt appears.

A.3 Special Operations for KED and KEX

You can use more than one KEX editor at the same time and you can print your text directly from KED or KEX without ending your work session.

A.3.1 Using Multiple KEX Editors

You can use more than one editor at the same time by creating multiple copies of KEX, opening the files you want to edit, and then moving from one file to another. Using auxiliary files, you can easily move file contents from a file opened on one editor to a file opened on another.

The number of KEX editors you can use is limited only by available memory or the 8-job limit of RT–11.

The KEX copies are independent. For example, the cursor position in one file does not change because of any editing you do in another file.

Since you are running the KEX copies as system jobs, while editing a file, you can get the monitor prompt (or another job) by pressing CTRL/B.

If your system has multiterminal support, you can run multiple copies of KEX, each from its own terminal, by using the SRUN and FRUN commands.

See Chapter 7 in the *Introduction to RT–11* for information and examples of using multiple KEX editors.

A.3.2 Printing Text Directly from KED or KEX Without Ending Your Work Session

You can use KED or KEX to print one or more sections of text without ending your work session.

If you run KEX as a system job, you must load the printer handler (LP or LS).

1. Make sure your printer is turned on.
2. Open an output channel to the printer handler (and your printer).
   a. Press GOLD and COMMAND
   b. Type `OPEN OUTPUT printer-handler:`

      `printer-handler` is LP or LS.
   c. Press ENTER
3. Select the section of text you want to print.
4. Copy that section to the printer handler (and your printer).
   a. Press GOLD and COMMAND
   b. Type `WRITE SELECT`
   c. Press ENTER

      Your printer prints some or all of the text you selected, depending on the size of the printer buffer and the amount of text you selected.
5. If you want to print additional sections of text, repeat steps 3 and 4. Otherwise, continue with the next step.

6. Close the output channel to the printer handler.

   a. Press GOLD and COMMAND

   b. Type `CLOSE`

   c. Press ENTER

      Your printer prints the rest of the text you selected (if any), and it may execute a form feed.

In other words:

Make sure your printer is turned on.
GOLD COMMAND `OPEN OUTPUT printer-handler:` ENTER
Select the section of text you want to print.
GOLD COMMAND `WRITE SELECT` ENTER
If necessary, repeat the two preceding steps.
GOLD COMMAND `CLOSE` ENTER

Appendix B
Command Line Options

This appendix describes the KED command line options.

B.1 Command Options

The following are the supported EDIT command options. Each option is followed by an operation description.

/COMMAND

The default, executes initialization file KEDINI.KED. If that file does not exist, KED does not report an error.

/COMMAND:init-filespec

Executes specified initialization file. If init-filespec does not exist, KED reports an error.

/JOURNAL

Produces a journal file that will let you recover your work after a system crash or similar malfunction. The file name of the journal file will be the name of your output file; the file type will be JOU.

/JOURNAL:jour-filespec

Specifies a different file specification for the journal file. You can type the file name or the file name and file type. If you type just the file name, the file type will be JOU.

/JOURNAL:jour-filespec/RECOVER:jour-filespec

Recovers the work you performed and produces another journal file that will contain the work you recovered and any additional work you might perform.

/JOURNAL/RECOVER

Same as above.

/NOCOMMAND

Suppresses initialization file processing.

/NOQUERY

Suppresses any queries that could be generated by commands in an initialization file.

/OUTPUT:filespec

Specifies a new file to contain the output of your work session. Keeps the original text in your input file.

/QUERY

The default, does not suppress queries that could be generated by commands in an initialization file.

/RECOVER

Uses the journal file with the default file specification to produce a text file that will contain the work you performed. You can then continue your work session or end it with a KED EXIT command.

/RECOVER:jour-filespec

Uses the journal file with the file specification you specified to produce a text file that will contain the work you performed.

You cannot use:

/JOURNAL with the /INSPECT or /READONLY file specification option

/OUTPUT with the /CREATE file specification option

/RECOVER with the /INSPECT or /READONLY file specification option

B.2 File Specification Options

The following are the supported EDIT command options for file specifications. Each option is followed by an operation description.

/ALLOCATE:size

Specifies a maximum output file size in number of blocks. The largest number allowed is 32767.

/CREATE

Creates a new file that you can edit.

/INSPECT

Lets you inspect, not edit, an existing file. You can use some of the KED functions and commands to move around inside that file and to create auxiliary output files.

/READONLY

Same as above.

You cannot use:

/CREATE with the /OUTPUT command option
/INSPECT with the /JOURNAL or /RECOVER command option
/READONLY with the /JOURNAL or /RECOVER command option

Appendix C
Functions and Commands

This appendix describes all the KED functions and commands you can use in a work session.

C.1 Functions

The *Bytes Required* column lists the number of bytes (characters) required in the macro buffer.

Key(s)	Bytes Required	Operation
↓	3	Advances the cursor to the corresponding character in the succeeding line. Enters your search model and searches forward without changing the keypad directional mode.
GOLD ↓	6	Moves the cursor to the end-of-file marker.
←	3	Backs up the cursor to the preceding character.
GOLD ←	6	Moves the cursor to the beginning of the current or preceding line.
→	3	Advances the cursor to the succeeding character.
GOLD →	6	Moves the cursor to a line terminator.
↑	3	Backs up the cursor to the corresponding character in the preceding line. Enters your search model and searches backward without changing the keypad directional mode.
GOLD ↑	6	Moves the cursor to the top of your file.
GOLD A	4	Aligns structured tabs according to the cursor's position and the indentation value you specified with the SET TABS command, thus setting the indentation level for TAB.
ADVANCE	3	Moves the cursor or sets it to move forward (right or down). Enters your search model, specifies your search direction, and starts the search.

Key(s)	Bytes Required	Operation
APPEND	3	Deletes selected text and stores it at the end of the paste buffer.
BACKUP	3	Moves the cursor or sets it to move backward (left or up). Enters your search model, specifies your search direction, and starts the search.
BLINE	3	Moves the cursor to the beginning of a line.
GOLD BOTTOM	6	Advances the cursor to the end of your file (the end-of-file marker).
CHAR	3	Moves the cursor one character at a time to the next or preceding character.
GOLD CHNGCASE	6	In selected text, changes all lowercase letters to uppercase and all uppercase letters to lowercase. If you did not select any text, changes the case of the cursor's character.
GOLD COMMAND	6	Lets you type a command on your keyboard.
CTRL/C	0	When your response to the `Model:` or `Command:` prompt is incomplete, cancels that prompt and restores the display. If the KED prompt appears, returns you to your system prompt.
CTRL/C CTRL/C	0	If KED is executing a function or command, cancels that operation and restores the display.
CTRL/J	1	Line-feed function. Deletes through the first character of the current or preceding word and stores the string in the word buffer.
GOLD CTRL/J	4	Inserts a deleted word from the word buffer back into your text.
CTRL/K	1	Inserts a vertical-tab character.
CTRL/L	1	Inserts a form-feed character.
CTRL/O	0	Interrupts output to your screen. To recover, press CTRL/O, and then, CTRL/R or CTRL/W.
CTRL/Q	0	Reenables output to your screen after a CTRL/S.

Key(s)	Bytes Required	Operation
CTRL/R	1	Restores the display after you press HELP or after the display is interrupted or corrupted for another reason. Restores the operation of the keypad.
CTRL/S	0	Interrupts output to your screen. To recover, press CTRL/Q.
CTRL/U	1	Deletes the line to the left of the cursor (but not the preceding line terminator) and stores the string in the line buffer. Corrects your responses to prompts.
GOLD CTRL/U	4	Inserts a deleted line from the line buffer back into your text.
CTRL/W	1	Same as CTRL/R.
CTRL/Z	0	When your response to the `Model:` or `Command:` prompt is incomplete, cancels that prompt and restores the display.
CUT	3	Deletes selected text and stores it in the paste buffer.
GOLD D	4	Decrements the indentation level counter for structured tabs, thus moving the structured tab column toward the left margin.
DELCHAR	3	Deletes the character to the right of the cursor and stores it in the character buffer.
GOLD DELEOL	6	Deletes the line to the right of the cursor (but not the line terminator) and stores it in the line buffer.
⌫	1	The delete key. Deletes the character to the left of the cursor and stores it in the character buffer. Corrects your responses to prompts.
GOLD ⌫	4	Inserts a deleted character from the character buffer back into your text.
DELLINE	3	Deletes the line to the right of the cursor (including the line terminator) and stores it in the line buffer.
DELWORD	3	Deletes the word to the right of the cursor and stores it in the word buffer.

Key(s)	Bytes Required	Operation
GOLD E	4	Increments the indentation level counter for structured tabs, thus moving the structured tab column toward the right margin.
ENTER	3	Executes the command you typed on your keyboard and erases that command and the `Command:` prompt. Restores the display after you press HELP. Enters your search model without starting the search. You must press FINDNEXT to start the search.
EOL	3	Moves the cursor to the end of a line (to a line terminator).
GOLD FILL	6	Reformats selected text lines to fit within the current right margin.
GOLD FIND	6	Displays the `Model:` prompt to let you specify a search string. You can use SET SEARCH commands to specify exact or general searching, the limit of your search, and the cursor's location at the search target and to disable the selection operation in the FIND function.
FINDNEXT	3	Uses the keypad directional mode to search for another occurrence of a target that matches your search model. Starts a search after you press ENTER.
GOLD	3	Selects the alternate function of a keypad key. Lets you repeat a function or use special character insertion.
HELP	3	If an error occurred, displays an error message. If no error occurred or if KED is displaying a message, displays a full-screen diagram of the keypad. If KED is displaying the keypad diagram, displays full-screen summaries of KED commands and functions.

Key(s)	Bytes Required	Operation
GOLD L	4	Forces the case of selected text or the cursor's character to lowercase.
LF	1	Line-feed key. Same as CTRL/J. Use only on VT100 compatible terminals.
GOLD LF	4	Same as GOLD CTRL/J. Use only on VT100 compatible terminals.
GOLD M identifier	5	Executes the macro specified by the identifier and marks it as the current macro.
N RETURN	0	Gives a negative answer to a prompt requiring a decision.
GOLD number	3+1/character in *number*	Repeats any functional operation, except CTRL/U, Delete, or SPECINS, the specified number of times Specifies the decimal ASCII equivalent of a nonprinting character.
GOLD OPENLINE	6	Breaks the current line by inserting a new-line terminator at the cursor's right.
PAGE	3	Moves the cursor to the beginning of a page. The page default is a form-feed character. To insert that character, press CTRL/L. You can also use the SET ENTITY PAGE command to redefine the beginning of a page.
GOLD PASTE	6	Inserts all the text you stored in the paste buffer.
GOLD REPLACE	6	Deletes and discards selected text, replacing text with paste buffer contents. Deletes and discards a search target and inserts a character string from the paste buffer.
GOLD RESET	6	Cancels a GOLD function or a text selection.
RETURN	2	Inserts a carriage return and line feed. After you type a command, executes the command, erases it, but continues to display the `Command:` prompt so you can type another command without pressing GOLD and COMMAND.
GOLD S	4	Ends the definition of a macro, marks it as the current macro, but does not execute it.

Key(s)	Bytes Required	Operation
SECTION	3	Moves the cursor to the beginning of a section. The section default is 16 lines. You can use the SET ENTITY SECTION command to redefine the beginning of a section.
SELECT	3	Starts a text selection operation by marking one end of the text. After moving the cursor to the other end, press APPEND, CHNGCASE, CUT, FILL, or REPLACE or use the FILL, TABS ADJUST, or WRITE SELECT command to process the text you selected. RESET cancels the select range.
GOLD SPECINS	6	Inserts a nonprinting character after you press GOLD and type the decimal ASCII equivalent of that character.
GOLD SUBSTITUTE	6	Substitutes a stored character string for a search target and then searches for the next occurrence of the search target. In other words, it completes the REPLACE function and then the FINDNEXT function.
TAB	1	Moves the cursor to the current indentation level.
GOLD TOP	6	Backs up the cursor to the beginning of your file.
GOLD U	4	Forces the case of selected text or the cursor's character to uppercase.
GOLD UNDELCHAR	6	Inserts a deleted character from the character buffer back into your text.
GOLD UNDELLINE	6	Inserts a deleted line from the line buffer back into your text.
GOLD UNDELWORD	6	Inserts a deleted word from the word buffer back into your text.
WORD	3	Moves the cursor one word at a time to the beginning of a word.

Key(s)	Bytes Required	Operation
GOLD X	4	If you are defining a macro, ends its definition, executes it, and marks it as the current macro. If you are not defining a macro, executes the current macro.
Y RETURN	0	Gives a positive answer to a prompt requiring a decision.

C.2 Equivalent Functions

The following functions perform equivalent operations but may require different amounts of (byte) memory in macro buffers. If you are concerned about memory availability in macro buffers, examine the following table to determine if one version of a function requires less memory than the other. Use the function that requires the fewest bytes.

Function	Bytes Required	Equivalent Function	Bytes Required
RETURN ←	5	GOLD OPENLINE	6
GOLD SUBSTITUTE	6	GOLD REPLACE FINDNEXT	12
SECTION	3	GOLD 16 BLINE	8
CTRL/R	1	CTRL/W	1
GOLD CTRL/U	4	GOLD UNDELLINE	6
GOLD ⌫	4	GOLD UNDELCHAR	6
GOLD LF[1]	4	GOLD UNDELWORD	6
LF[1]	1	CTRL/J	1
GOLD CTRL/J	4	GOLD UNDELWORD	6
GOLD ↓	6	GOLD BOTTOM	6
GOLD ←	6	BACKUP BLINE	6
GOLD →	6	ADVANCE EOL	6
GOLD ↑	6	GOLD TOP	6
←	3	BACKUP CHAR	6
→	3	ADVANCE CHAR	6
RETURN GOLD OPENLINE	8	GOLD OPENLINE GOLD OPENLINE	12

[1] Use only on VT100 compatible terminals.

Function	Bytes Required	Equivalent Function	Bytes Required
DELLINE ⌫	4	DELLINE DELLINE	6
ESC ESC	2	GOLD 27 GOLD SPECINS	11

C.3 Commands

Square brackets ([ ]) enclose the parts you do not have to type.

When included in a macro buffer, each command requires 9 bytes to summon the command prompt and enter the command and 1 byte for each character in the actual command string. Therefore, in macros, you should use the shortest valid abbreviation for command strings. Also, when issuing multiple commands, if you execute commands with RETURN instead of ENTER, you save 7 bytes for each command.

Command	Operation
@filespec	Executes your macro directly from an auxiliary file.
CH[ANGE]	Leaves the command mode and returns KED to the text-editing mode. In an auxiliary command file, it can precede text that you want to insert into your text file.
CLE[AR] A[LL MACROS]	Deletes all macros from the macro buffer.
CLE[AR] M[ACRO] identifier	Deletes the macro specified by the identifier but does not delete any other macros from the macro buffer.
CLE[AR] P[ASTE]	Empties the paste buffer (deletes its contents).
CLO[SE]	Closes your auxiliary output file.
EXIT	Ends your work session, closes (saves) all open files, and displays your system prompt.
EXIT S[AVE]	Ends your work session and saves the journal file.

Command	Operation
F[ILL]	Reformats selected text to fit within the current right margin. The default line length is 78 characters when the screen width is 80 and 130 characters when the screen width is 132.
INC[LUDE] number [LINES]	Inserts the specified number of lines from your auxiliary input file.
INC[LUDE] number P[AGES]	Inserts the specified number of pages from your auxiliary input file, according to the default page definition (the form-feed character) or the latest definition you specified with a SET ENTITY PAGE command.
INC[LUDE] R[EST]	Inserts all the text remaining in your auxiliary input file.
LE[ARN]	Enables the storing and execution of your sequence of functions and commands (your macro). KED assigns the backslash character as your identifier.
LE[ARN] identifier	Same as above, except you specify the identifier, which can be any single letter or the backslash character.
LOA[D] identifier	Copies an auxiliary file to the macro buffer. The identifier can be any single letter that KED is not using as a macro identifier. The file name will be your identifier and the file type will be KED.
LOA[D] identifier filespec	Same as above, except you specify a file specification for the auxiliary file. If you omit the file type, it will be KED.
LOC[AL]	Renumbers your MACRO–11 local symbols, using a default of 10 for each of the missing parameters.
LOC[AL] start	Renumbers your MACRO–11 local symbols, using a default of 10 for the missing increment parameter.

Command	Operation
LOC[AL] start increment	Renumbers your MACRO–11 local symbols so they start with the starting value and are separated by the increment. The default starting value is 10; the default increment is also 10.
[OPEN] I[NPUT] filespec	Opens your auxiliary input file.
[OPEN] O[UTPUT] filespec	Opens your auxiliary output file.
PURGE	Deletes your auxiliary output file.
QUIT	Ends your work session, deletes all open output files, and displays your system prompt.
QUIT S[AVE]	Ends your work session and saves the journal file.
SA[VE] identifier	Copies your macro to an auxiliary file. The identifier must be the same identifier you used in creating and storing the macro; it cannot be the backslash character. The file name will be your identifier and the file type will be KED.
SA[VE] identifier filespec	Same as above, except you specify a file specification for your macro. If you omit the file type, it will be KED.
SE[T ENTITY] P[AGE] "marker"	Defines a page to be all the text between occurrences of the specified marker. The form-feed character is the default marker.
SE[T ENTITY] P[AGE] "marker"	Same as above.
SE[T ENTITY] P[AGE] number [LINES]	Defines a page to be the specified number of lines.
SE[T ENTITY] SEC[TION] "marker"	Defines a section to be all the text between occurrences of the specified marker.
SE[T ENTITY] SEC[TION] "marker"	Same as above.
SE[T ENTITY] SEC[TION] number [LINES]	Defines a section to be the specified number of lines. The default is 16 lines.
SE[T] F[REQUENCY] number	Changes the modification count that determines when the contents of the journal buffer will be moved to the journal file.

Command	Operation
SE[T] H[ELP]	The default. Enables the display of the keypad diagram and the other full-screen HELP displays.
SE[T] J[OURNAL] S[AVE]	Saves the journal file when you end your work session with an EXIT or QUIT command.
SE[T] NOH[ELP]	Suppresses all full-screen HELP displays.
SE[T] NOQ[UIET]	The default. Sets the terminal warning signal to be the sound of a beeper.
SE[T] NOT[ABS]	Cancels the structured tab operation (your indentation value and the initialization of the level counter).
SE[T] NOW[RAP]	Cancels your right margin setting (your maximum line length). It cancels the word-wrap operation but does not affect the FILL function.
SE[T] Q[UIET]	Sets the terminal warning signal to be a reversal of the text and background.
SE[T SCREEN] 80	The default. Displays up to 78 characters (columns) in a single line.
SE[T SCREEN] 132	Displays up to 130 characters (columns) in a single line.
SE[T SCREEN] D[ARK]	The default. Displays light text on a dark background.
SE[T SCREEN] L[IGHT]	Displays dark text on a light background.
SE[T SEARCH] BE[GIN]	The default. Specifies that the cursor will be placed at the beginning of the matching target.
SE[T SEARCH] BO[UNDED]	Limits searches to one page, according to the default definition (the form-feed character) or the latest definition you specified with a SET ENTITY PAGE command.
SE[T SEARCH] EN[D]	Specifies that the cursor will be placed at the end of the matching target.

Command	Operation
SE[T SEARCH] EX[ACT]	Specifies that the target and model letters will match with regard for case (distinguish between uppercase and lowercase letters in your search string).
SE[T SEARCH] G[ENERAL]	The default. Specifies that the target and model letters will match without regard for case (does not distinguish between uppercase and lowercase letters in your search string).
SE[T SEARCH] NOR[ETURN]	Specifies that the cursor will move to the top or bottom of your file when a matching target is not found.
SE[T SEARCH] NOSEL[ECT]	The default. The search target is not used as the selection for APPEND, CHNGCASE, and CUT functions (disables the selection operation in the FIND function).
SE[T SEARCH] R[ETURN]	The default. Specifies that the cursor will remain in its current location when a matching target is not found.
SE[T SEARCH] SEL[ECT]	Lets the search target be used as the selection for APPEND, CHNGCASE, and CUT functions (enables the selection operation in the FIND function).
SE[T SEARCH] U[NBOUNDED]	The default. Allows searches for a matching target between the cursor and the top or bottom of your file.
SE[T] T[ABS]	For the structured tab operation, specifies a default indentation value of 4 and initializes the level counter at 1.
SE[T] T[ABS] indent	Same as above, except you specify the indentation value.
SE[T] W[RAP]	Sets your right margin (your maximum line length) for the word-wrap operation and the FILL function to the default, which is 78 characters when the screen width is 80 and 130 characters when the screen width is 132.

Command	Operation
SE[T] W[RAP] number	Same as above, except you set the right margin instead of using the default.
SK[IP] number [LINES]	Skips the specified number of lines in your auxiliary input file.
SK[IP] number P[AGES]	Skips the specified number of pages in your auxiliary input file, according to the default page definition (the form-feed character) or the latest definition you specified with a SET ENTITY PAGE command.
SK[IP] R[EST]	Skips all the text remaining in your auxiliary input file.
[TABS] A[DJUST]	Changes the indentation level of lines in selected text to the default, which rewrites indentations to use the most efficient combination of tabs and spaces.
[TABS] A[DJUST] 0	Same as above.
[TABS] A[DJUST] level	Same as above, except you specify the indentation level.
W[RITE] J[OURNAL]	Moves the contents of the journal buffer to the journal file without waiting for the modification count to be reached.
W[RITE] number [LINES]	Copies the specified number of lines to your auxiliary output file.
W[RITE] number P[AGES]	Copies the specified number of pages to your auxiliary output file, according to the default page definition (the form-feed character) or the latest definition you specified with a SET ENTITY PAGE command.
W[RITE] R[EST]	Copies all the text (from the cursor to the end-of-file marker) to your auxiliary output file.
W[RITE] S[ELECT]	Copies selected text to your auxiliary output file.

Appendix D

ASCII Decimal Equivalents

This appendix lists the decimal equivalents of the nonprinting ASCII characters. The nonprinting characters you can use in a KED work session depends on your operating system.

See Section 10.1 for information on inserting nonprinting characters in your text file.

ASCII Character	Decimal Equivalent	KED Operation in Text File
NUL	0	Invalid
SOH	1	Inserts CTRL/A
STX	2	Inserts CTRL/B
ETX	3	Inserts CTRL/C
EOT	4	Inserts CTRL/D
ENQ	5	Inserts CTRL/E
ACK	6	Inserts CTRL/F
BEL	7	Inserts CTRL/G
BS	8	Inserts CTRL/H
HT	9	Same as Tab, inserts horizontal tab
LF	10	Inserts line feed
VT	11	Same as CTRL/K, inserts vertical tab
FF	12	Same as CTRL/L, inserts form feed
CR	13	Inserts carriage return
SO	14	Inserts CTRL/N
SI	15	Inserts CTRL/O
DLE	16	Inserts CTRL/P
DC1	17	Inserts CTRL/Q
DC2	18	Inserts CTRL/R
DC3	19	Inserts CTRL/S
DC4	20	Inserts CTRL/T

ASCII Character	Decimal Equivalent	KED Operation in Text File
NAK	21	Inserts CTRL/U
SYN	22	Inserts CTRL/V
ETB	23	Inserts CTRL/W
CAN	24	Inserts CTRL/X
EM	25	Inserts CTRL/Y
SUB	26	Inserts CTRL/Z
ESC	27	Inserts π (CTRL/[)
FS	28	Inserts ^\
GS	29	Inserts ^]
RS	30	Inserts ^^
US	31	Inserts CTRL/_
SP	32	Inserts space character
DEL	127	Inserts delete character

Appendix E
Work Session Example

This appendix demonstrates some of the most common KED functions, thus giving you an example of KED in action. This appendix also:

- Gives you a good introduction to the keypad keys
- Shows you what your terminal screen looks like while you are editing
- Illustrates some of the ways that KED changes your terminal screen
- Lets you see what the KED cursor and end-of-file symbol look like on your system
- Proves that the names of the KED functions explain most of what the functions do

E.1 Requirements

To work through the example in this appendix, you need:

A copy of the keypad diagram
The file DEMOKD.TXT

You will find the keypad diagram in Chapter 2 and on the front of the *PDP–11 Keypad Editor Reference Card*.

DEMOKD.TXT is on one of your distribution volumes, and it may be on your system volume or on one of your working volumes.

If DEMOKD.TXT is on your system volume, you can leave it there or copy it to one of your working volumes, depending on the size of your system volume. You will need approximately 18 blocks of contiguous free space (4 blocks for DEMOKD.TXT, 14 blocks for your output file) to complete the example.

If DEMOKD.TXT is not on your system volume or on one of your working volumes:

1. Install the distribution volume containing DEMOKD.TXT in an available drive.
2. Copy DEMOKD.TXT to your system volume or to one of your working volumes.

In any event, unprotect DEMOKD.TXT.

E.2 Example

The first time you work with this example, follow the instructions carefully because each step depends on the preceding step. Do not experiment while working with the example for the first time.

For each step in the example:

Read the instruction.
Look at your terminal screen while typing.
Watch how KED performs the operation you specified.
Read the short description that follows the instruction.

Your terminal may use the block cursor. Displays in this example use the underline cursor.

1. Begin your work session, specifying DEMOKD.TXT as the file you want to edit.

   Type the following command, where dev is your default data volume (DK) or one of your working volumes.

   ```
   .EDIT [dev:]DEMOKD.TXT RET
   ```

   KED displays the first 24 lines in DEMOKD.TXT.

   ```
    Sample Keypad Editing Session - //DATE//

        This file has been designed especially for the sample editing session
   that is described in the PDP-11 Keypad Editor User's Guide.

   After you have completed the sample session,
   you may use this file to practice other keypad editor functions and
   commands, if you like.

   ABOUT THE SAMPLE SESSION

   Your general task for the sample session is to insert the date you
   begin working with the keypad editor into your own copy of this file. The
   date is //DATE// .

   Sample Keypad Editing Session, "section" 2 lakdjhfuierbhnv a678237yfg^90(4#!!


   The PDP-11 Keypad Editor is for use with
   VT100 compatible terminals that have keypads.
   ```

   You will see the remaining lines in DEMOKD.TXT as you progress through this work session.

2. Type your name. Use the keyboard keys exactly as if you were using an electric typewriter. To correct your typing mistakes, press the ⌫ (delete) key as many times as necessary and then type the correct characters.

   KED displays each character (letter or space) as you type it.

3. Use the end-of-line (EOL) function. Press the 2 key on your keypad.

   KED moves the cursor to the end of the first line.

4. Use the delete-left-word (line feed) function. Press the LF key on your keyboard (CTRL/J on some keyboards).

   KED deletes the word at the cursor's left. (KED also stores that word, in case you want to recover it. A later step shows you how to recover a deletion.)

5. Type the current date. Use the number and letter keys on your keyboard. To correct your typing mistakes, press the ⌫ key as many times as necessary and then type the correct characters.

   KED displays each character (number, letter, or space) as you type it.

6. Use the end-of-section (SECTION) function. Press the 8 key on your keypad.

   KED moves the cursor 16 lines forward. (KED's default definition of a section is 16 text lines.)

7. Use the beginning-of-word (WORD) function three times. Press the 1 key on your keypad three times.

   For each WORD function, KED moves the cursor to the beginning of a word.

8. Use the delete-right-line (DELLINE) function. Press the PF4 key on your keypad.

   KED deletes the cursor's character, the characters to the right of the cursor, and the line terminator. Then it restores your screen to show the effect of the deletion. (KED also stores all the deleted characters and the line terminator, in case you want to recover them. The next step shows you how to recover a line deletion.)

9. The preceding step deleted too much. Therefore, use the undelete-line (UNDELLINE) function to recover the entire deletion. Press the PF1 key and then the PF4 key on your keypad.

   KED restores all the characters you deleted, including the line terminator.

   Most of the keys on your keypad specify two KED functions: a standard function and an alternate function. The functions shown on the top of the keys in the keypad diagram are standard functions. The functions shown on the bottom of the keys are alternate functions. You specify a standard function by pressing the appropriate key. You specify its alternate function by first pressing the select-alternate (GOLD) PF1 key.

   On the PF4 key, DELLINE is the standard function. Unless you press the PF1 key first, that is the function KED will perform when you press the PF4 key. The alternate function of the PF4 key restores the last line you deleted with DELLINE or one of KED's other line-deletion functions. To use that alternate function, you must use the select-alternate function first (press the PF1 key) and then use the UNDELLINE function (press the PF4 key).

10. Use the beginning-of-word (WORD) function two more times. Press the 1 key on your keypad two more times.

    KED moves the cursor two words forward.

11. Use the delete-to-end-of-line (DELEOL) function. Press the GOLD (PF1) key and then the DELEOL (2) key on your keypad.

    KED deletes the right-hand part of the line and stores the deletion. It does not delete the line terminator.

12. Use the move-to-end-of-file (BOTTOM) function. Press the GOLD (PF1) key and then the BOTTOM (4) key on your keypad.

    KED moves the cursor past all the characters in your file to the end-of-file marker.

13. KED's search functions can move the cursor directly to any character string you specify. As shown below, you specify the FIND function, complete that function with a model (a search target), and then specify the direction of your search.

    a. Use the specify-search-target (FIND) function. Press the GOLD (PF1) key and then the FIND (PF3) key on your keypad.

       KED temporarily erases the top two screen lines and displays the prompt `Model:`.

    b. Type the search string `//DATE//` (two slashes, the word DATE, and two more slashes).

       KED displays each character of the model (your search target) as you type it.

    c. Use the move-backward (BACKUP) function to specify a backward search. Press the BACKUP (5) key on your keypad.

       KED searches backward for the nearest occurrence of a character string that matches your model. When it finds a matching string, it places the cursor on the first character in that string.

14. Use the delete-right-word (DELWORD) function and type the current date again. Press the DELWORD (-) key on your keypad and then type the current date (and a period).

    KED deletes the next word, stores the characters you deleted, and then displays each new character as you type it.

15. Use the continue-search (FINDNEXT) function. Press the FINDNEXT (PF3) key on your keypad.

    KED uses the latest model you specified (`//DATE//`) in searching backwards for another occurrence of a matching string. KED searches backward because that is the direction you specified most recently. The cursor is now on one of the three remaining occurrences of `//DATE//`.

16. This step gives you an easier way to change the remaining occurrences of `//DATE//` to the current date. You will use the SUBSTITUTE function, but since that function uses KED's paste buffer to make substitutions, the first thing you must do is put your substitution string in the paste buffer.

    a. Use the begin-selection (SELECT) function. Press the SELECT (.) key on your keypad.

       The SELECT function begins one end of a string to be selected. Your screen does not show any effects of selecting until you move the cursor.

b. Type the current date again.

   As you type the characters, KED underlines them or displays them in reverse video form.

c. Use the cut-out-selection (CUT) function. Press the CUT (6) key on your keypad.

   The CUT function deletes all the characters in your selection and stores them in the paste buffer. The cursor is still on the first character of `//DATE//`.

d. Use the substitute-and-search (SUBSTITUTE) function. Press the GOLD (PF1) key and then the SUBSTITUTE (Enter) key on your keypad.

   KED substitutes the current date (in its paste buffer) for your search target (`//DATE//`) and then executes the FINDNEXT function.

17. Use the substitute-and-search (SUBSTITUTE) function again. Press the GOLD (PF1) key and then the SUBSTITUTE (Enter) key on your keypad.

    Each time you use the SUBSTITUTE function, KED substitutes and then searches. In this step, KED makes the substitution but fails to find another string that matches your model. KED signals the failure by ringing a bell, sounding a beeper, or reversing the text and background on your terminal.

18. Use the HELP function. Press the HELP (PF2) key on your keypad.

    When the SUBSTITUTE function failed, KED placed an explanation of the failure in a message buffer. In this step, the HELP function temporarily erases the bottom three lines on your screen and then displays that explanation.

19. Use the restore-screen (CTRL/W) function. Press the CTRL/W keys on your keyboard (hold the CTRL key down while you type the letter W).

    KED deletes the HELP message and restores the three lines it deleted.

20. Use the EXIT command to end your work session and save your work in a new version of DEMOKD.TXT.

    a. Press the GOLD (PF1) key and then the COMMAND (7) key on your keypad.

       KED temporarily erases the top two screen lines and displays the prompt `Command:`.

    b. Type the word EXIT on your keyboard.

       KED displays each letter of the command as you type it.

    c. Press the Enter key on your keypad.

       KED moves the cursor to the lower left corner of your screen, pauses briefly, and then returns you to your operating system prompt.

21. If KED displays its asterisk (*) prompt instead of returning you to your operating system prompt, press CTRL/C key on your keyboard.

    CTRL/C returns you to the operating system prompt.

Now, the date you typed.

```
Ed Bruce Sample Keypad Editing Session - 22 Sep 89

     This file has been designed especially for the sample editing session
that is described in the PDP-11 Keypad Editor User’s Guide.

After you have completed the sample session,
you may use this file to practice other keypad editor functions and
commands, if you like.

ABOUT THE SAMPLE SESSION

Your general task for the sample session is to insert the date you
begin working with the keypad editor into your own copy of this file. The
date is 22 Sep 89.


Sample Keypad Editing Session, "section"



The PDP-11 Keypad Editor is for use with
VT100 compatible terminals that have keypads.

As you work through a file with the keypad editor, the editor displays 24
lines of the file on your screen. One of the lines always shows the keypad
editor cursor, the flashing symbol that is always somewhere on your screen.
The cursor identifies the character that will be erased if you use one of the
erasing functions. It also marks the location in your file where the editor
will insert what you type on the keyboard.

22 Sep 89 is the date you started to learn to use the keypad editor.
If KED is the first video-oriented editor you have used, welcome to a
new world of fast, clear responses to editing functions. If you already
have used a video-oriented editor, welcome to Digital’s contribution
for users of the RT-11 operating system. If the
experience of other users is valid for you also, you will be confident about
using the keypad editor within a day or two after 22 Sep 89.
```

Appendix F
Messages

This appendix describes the KED messages you can encounter during a work session.

- Section F.1 describes the different types of messages.
- Section F.2 tells you what to do if you ever encounter a nonrecoverable condition.
- Section F.3 is for RT–11 users. It tells you what to do when an output file is full.
- Section F.4 explains how you can accidentally create a lot of extraneous characters on your terminal screen.
- Section F.5 lists, explains, and gives corrective actions for all the messages that KED can display.

F.1 Message Types

KED displays three types of messages:

Command level messages
HELP function messages
Prompts and informative messages

F.1.1 Command Level Messages

Command level messages appear when your response to a system or KED prompt contains or causes an error. Such messages have the following general form.

```
?KED-Severity code-Text
```

or

```
?KEX-Severity code-Text
```

You will see only two severity codes, F or W:

Severity Code	**Meaning and KED Operation**
F	A fatal error. KED returns to your system prompt or to the KED utility prompt.
W	A warning. KED continues executing unless you stop it.

KEX is a virtual KED program; the severity code and text for KEX and KED messages are the same. If you see a `?KEX-Severity code-text` message, look it up in Section F.5 under `?KED-Severity code-text`.

F.1.2 HELP Function Messages

When KED is displaying a file and cannot complete your function or command, it signals you by sounding a beeper or reversing the text and background on your terminal.

If that happens:

1. Try to determine the cause of the signal by inspecting the cursor's position or the changes that are displayed on your screen.
2. If you need more information, press HELP to display a 1-line message (centered on the bottom of your screen) that will explain the signal.

   You can return to your file by pressing ENTER or CTRL/W.

F.1.3 Prompts and Informative Messages

When KED is displaying a file, as the need arises, it also displays prompts and informative messages on the top line of your screen.

KED restores your screen when it completes an operation that is related to the prompt or informative message. For example, when you move the cursor from the beginning to the end of a very long file, KED displays `WORKING...` on the top screen line. When the cursor reaches the end of that file, KED removes the message and restores your screen.

F.2 Nonrecoverable Conditions

Some of the messages displayed by KED identify nonrecoverable conditions. Digital uses those messages to diagnose KED malfunctions.

If you encounter a nonrecoverable condition:

1. As accurately as possible, write down the message, the functions, and the commands you used before the message appeared.
2. Save all the files you were editing (input and output files).
3. Obtain a new copy of the appropriate version of KED from your distribution kit.
4. Using the new copy of KED, try to duplicate the functions and commands that caused the message.
5. If the message appears again, make sure your hardware is connected and functioning properly.
6. If your hardware is connected and functioning properly and you qualify to receive a written reply under Digital's Software Performance Report (SPR) service, follow the instructions on the SPR form.

F.3 When an RT–11 Output File Is Full

Under RT–11, KED displays a message when the file you are editing or creating becomes too large to allow more insertions during your work session.

When that happens, try one or more of the following procedures.

- Exit and begin another work session.

  KED opens a new output file when you begin another work session. In most cases, the new output file will be large enough to hold the additions you want to make.

  That procedure may not be effective if your output volume is almost full or if free space on it is not consolidated.

  — If free space on your output volume is not consolidated, use the SQUEEZE command.

  — If your output volume is almost full, use the next procedure or use a new output volume.

  The principal disadvantage of the procedure is that the EXIT command:

  Closes all open files
  Discards the contents of the paste, line, word, and character buffers
  Restores KED's default settings

- Delete and make room for insertions.

  When your output file is full and valuable text is in the paste buffer or the other buffers, you can make room for the text by deleting part of your file. Each character you delete makes room for one additional character during your work session.

- Use an auxiliary output file.

  When your output file is full, you can split it into two files.

  1. Use the OPEN OUTPUT command to open an auxiliary output file and the WRITE command to store as much of your output file as necessary.
  2. Use the CLOSE command to save the auxiliary output file and then delete the section of your output file that you stored in the auxiliary output file.

  In most cases, you can recombine the parts of your original file in another work session. If free space is still not sufficient, use a new output volume.

F.4 When KED Seems to Insert Extraneous Characters

Under rare circumstances, hardware conditions can cause extraneous characters (characters you did not type) such as [A, [B, [C, [D, or O, to appear in your text file. They appear because an overfilled buffer disrupts the stream of characters being sent to KED, and KED interprets some of them as insertions, not editing operations.

To prevent the problem, avoid unusually long bursts of automatic repeating.

To correct the problem, use keypad functions to delete the unwanted characters or use the QUIT command to end your work session without creating an output file.

F.5 Messages

These messages are arranged alphabetically (according to the first letter and later characters in each message).

If you are looking for a ?KEX message, look it up as a ?KED message.

Advance char finds end of file

Explanation: The CHAR or right-arrow function moved the cursor to the bottom of your file.

Your Response: Use any valid function or command.

Advance line finds end of file

Explanation: The BLINE function moved the cursor to the end of your file. With line count definitions, the PAGE and SECTION functions can also cause that message.

Your Response: Use another valid function.

Advance word finds end of file

Explanation: The WORD function moved the cursor to the bottom of your file or the DELWORD function deleted the last character in your file.

Your Response: Use any valid function or command.

All available learn macro buffers are active

Explanation: There are no learn buffers left to use.

Your Response: Clear a learn buffer and use it.

Argument error to INCLUDE or SKIP

Explanation: KED did not complete an INCLUDE or SKIP command because you specified a negative line or page count.

Your Response: Use any valid function or command.

Argument error to WRITE

Explanation: KED did not complete a WRITE command because you specified a negative line or page count.

Your Response: Use any valid function or command.

Arrow command finds extremity of file

Explanation: The up-arrow function moved the cursor to the top of your file, or the down-arrow function moved the cursor to the bottom of your file.

Your Response: Use any valid function or command.

Auxiliary input file not open

Explanation: KED did not complete your INCLUDE command because the auxiliary input file is not open. KED closes an auxiliary input file when an INCLUDE or SKIP command processes the entire file.

Your Response: To open an auxiliary input file, use the OPEN INPUT command. Otherwise, use any valid function or command.

Auxiliary output error

Explanation: KED was unable to write to the auxiliary output file, probably due to an I/O error.

Your Response: Be sure the device on which the auxiliary output file resides is available and on line. If KEX is being run as a system job, be sure the device handler for the device is loaded.

Auxiliary output file exists

Explanation: You answered N to the following message.

Your Response: Use any valid function or command.

Auxiliary output file exists—Replace (Y,N)?

Explanation: The auxiliary output file you specified has the same name as an existing file on your output volume.

Your Response: To preserve the existing file, type N and press RETURN.

To delete the existing file and create a new file with the same name, type Y and press RETURN.

Auxiliary output file full

Explanation: No more space remains in your auxiliary output file.

Your Response: Use the CLOSE command to close your auxiliary output file. Then, open another file for auxiliary output, and merge the files in another work session. Use the COPY command or the PIP utility to merge the files.

Auxiliary output file not open

Explanation: KED did not complete your WRITE command because an auxiliary output file is not open. The CLOSE and EXIT commands close an auxiliary output file. The PURGE and QUIT commands discard an auxiliary output file if one is open.

Your Response: To open an auxiliary output file, use the OPEN OUTPUT command. Otherwise, use any valid function or command.

Backup char finds beginning of file

Explanation: The CHAR or left-arrow function moved the cursor to the top of your file.

Your Response: Use any valid function or command.

Backup line finds beginning of file

Explanation: The BLINE function moved the cursor to the top of your file. With line count definitions, the PAGE and SECTION functions can also cause that message.

Your Response: Use another valid function.

Backup word finds beginning of file

Explanation: A WORD or line-feed function moved the cursor to the top of your file.

Your Response: Use any valid function or command.

Bounded search reached bound

Explanation: You used a SET SEARCH BOUNDED command to limit searches to one page. Then, a FIND or FINDNEXT function moved the cursor to the top or bottom of a page without finding a character string that matches the model you specified.

Your Response: Use any valid function or command.

CHNGCASE finds end of file

Explanation: In the ADVANCE mode, the CHNGCASE function changed the case of the last character in your file.

Your Response: Use another valid function or command.

Command:

Explanation: You pressed GOLD and COMMAND.

Your Response: Type the command you want KED to execute and then press ENTER.

Command canceled

Explanation: You pressed CTRL/Z or CTRL/C to cancel a command and the command prompt.

Your Response: Use any valid function or command.

CTRL/C entered to stop operation

Explanation: You pressed CTRL/C to cancel a command or function. KED restores your screen and shows where the cursor moved. If you typed commands or functions while KED was displaying the `WORKING...` message, KED ignores all of them.

Your Response: Use any valid function or command.

CTRL/C or CTRL/Z ignored—use QUIT

Explanation: While you are working in a file (no prompts are displayed, no operation is being performed), KED ignores CTRL/C and CTRL/Z.

Your Response: To insert CTRL/C or CTRL/Z in your file, use the SPECINS function.

To end your work session without saving any open output files, use the QUIT command.

CTRL/U finds beginning of file

Explanation: The CTRL/U function deleted the first character in your file.

Your Response: Use any valid function or command.

Cursor not at target

Explanation: With no selected text, KED did not complete a CUT, APPEND, PASTE, or SUBSTITUTE function, because the cursor is not at a valid search target.

Your Response: Use any valid function or command.

DELCHAR finds end of file

Explanation: The DELCHAR function deleted the last character in your file.

Your Response: Use any valid function or command.

DELEOL finds end of file

Explanation: The DELEOL function deleted the last character in your file.

Your Response: Use any valid function or command.

DELETE finds beginning of file

Explanation: The Delete function deleted the first character in your file.

Your Response: Use any valid function or command.

DELLINE finds end of file

Explanation: The DELLINE function deleted the last character in your file.

Your Response: Use any valid function or command.

Empty select range specified to CHNGCASE

Explanation: After using the SELECT function, you used the CHNGCASE function or pressed CTRL/L or CTRL/U before selecting text.

Your Response: Press SELECT, move the cursor to select the letters you want to change, and then use the CHNGCASE, GOLD L, or GOLD U function.

End of file reached on auxiliary input file

Explanation: A SKIP or INCLUDE command read to the bottom of your auxiliary input file, and KED closed that file.

Your Response: Use any valid function or command.

EOL finds beginning of file

Explanation: The EOL function moved the cursor to the top of your file.

Your Response: Use any valid function or command.

EOL finds end of file

Explanation: The EOL function moved the cursor to the bottom of your file.

Your Response: Use any valid function or command.

Error reading auxiliary input file

Explanation: KED did not read the auxiliary input device. The problem is not due to reaching the end of the file on that device.

Your Response: If the file name was correct, check that the device is on line and contains the proper volume. Use another device if the device is faulty.

Error writing auxiliary output or journal file

Explanation: KED may have detected a bad block or a faulty device.

Your Response: Check that your device is on line and write enabled. Check that your volume is properly formatted. Try using another auxiliary output device or volume. Use the DIRECTORY/BADBLOCKS command to check for bad blocks.

Error un/nesting input modes

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

EXIT stopped by CTRL/C

Explanation: You pressed CTRL/C twice to cancel an EXIT command. The results are unpredictable because they depend on:

Your output file size
The speed of your hardware components

When you pressed the last CTRL/C

Your Response: If KED displays its command prompt again, you were too late in canceling the EXIT process. In that case, your output files are saved. If you specified automatic backup of your input files, they are renamed.

If KED restores your screen with the part of your input file that contains the cursor, you were not too late in canceling the EXIT process. In that case, continue editing by using any valid function or command.

File full during FILL

Explanation: KED did not complete your FILL command or function because the output file or temporary file is full.

Your Response: See Section F.3.

File full during INCLUDE

Explanation: KED did not complete your INCLUDE command because your output file is not large enough for the text you want to copy. The cursor is to the right of the last character KED copied from your auxiliary input file.

Your Response: See Section F.3.

You can also use the CLOSE or QUIT command and any functions or commands that delete text from your input and output files.

INCLUDE finds end of auxiliary input file

Explanation: Your INCLUDE command copied the last character in the auxiliary input file to your document file. KED closed the auxiliary input file.

Your Response: Use another valid function or command.

Insert failure in LOCAL

Explanation: You pressed CTRL/C to stop the processing of a LOCAL command, there is not enough space in the output file, or KED did not finish processing for some other reason.

Your Response: End your work session, begin it again and try another LOCAL command. Correct any erroneous local symbols in the current LSB. Delete any ^A and ^D markers left by the LOCAL process.

Insert finds file full

Explanation: KED did not insert the character you typed because your output file is not large enough.

Your Response: See Section F.3.

You can also use the CLOSE or QUIT command or any functions or commands that delete text from the file.

Invalid command

Explanation: You entered a command that KED could not accept.

Your Response: Use any valid function or command.

Invalid definition of PAGE or SECTION

Explanation: KED did not process your SET PAGE or SET SECTION command for one of the following reasons.

Your marker was not enclosed by single quotes or double quotes.
Your marker was enclosed by invalid delimiters.
In your line count definition, you did not specify a value greater than zero.

Your Response: Use any valid function or command.

Invalid file specification

Explanation: The file specification you typed in an OPEN INPUT or OPEN OUTPUT command was not valid.

Your Response: Reenter the command, using a valid file specification.

Invalid frequency value

Explanation: You entered an invalid number for the SET FREQUENCY command. Valid numbers are 0–511 (0 turns off frequency limit).

Explanation: Repeat the command and enter a valid frequency number.

Invalid function

Explanation: KED did not recognize the function you used and did not insert the characters you typed. The most common cause is pressing ESC (escape) and then typing almost any other key(s) on your keyboard or keypad.

Your Response: To insert an escape character or another character you cannot insert directly, use the SPECINS function.

You can also use any valid function or command.

Invalid learn macro name

Explanation: You attempted to assign a name of other than one alphabetic character to a learn macro.

Your Response: Assign only a single alphabetic character as a learn buffer name.

Invalid nesting of learn commands

Explanation: You attempted to nest learn commands, which is not allowed.

Your Response: Terminate the first learn command before issuing another.

Invalid parameter to ADJUST

Explanation: In your TABS ADJUST command, you specified an adjustment that is outside the range of -50 to +50.

Your Response: Reenter the command, using the correct adjustment, or use another valid command or function.

Invalid parameter to LOCAL

Explanation: In your LOCAL command, you specified a starting value or increment that is 0 or greater than 32767.

Your Response: Reenter the command, using the correct starting value or increment, or use another valid command or function.

Invalid right column for wrap

Explanation: You specified a right margin greater than 256 in your SET WRAP command.

Your Response: Reenter the command, using the correct right margin, or use another valid function or command.

Invalid tab indent value

Explanation: You specified an indentation value of 0 or less in your SET TABS command.

Your Response: Reenter the command, using the correct indentation value, or use another valid function or command.

Invalid terminating key

Explanation: KED did not complete your FIND function because:

You included an invalid character in your model.
or
You did not press ADVANCE or BACKUP to enter your model.

Your Response: Use any valid function or command.

Invalid terminating key to command prompt

Explanation: KED did not process your command because:

You included an invalid character in your command.
or
You did not press ENTER after typing your command.

Your Response: Use any valid function or command.

Invalid to insert null

Explanation: You typed GOLD 0 GOLD SPECINS, hoping to insert a null character.

Your Response: Do not try to insert a null character. Use another valid function or command.

I/O error closing auxiliary output file

Explanation: KED did not write to your auxiliary output file during a CLOSE operation. KED may have detected a bad block.

Your Response: Try using another auxiliary output volume. Use the DIRECTORY /BADBLOCKS command to check for bad blocks.

Journal output file full. Journaling terminated.

Explanation: No more room remains in the journal output file.

Your Response: Exit from the work session and start a new work session if you wish to continue journal file protection. You can continue with the current work session but will not have journal file protection.

Journal file output error—Exist session (Y/N)?

Explanation: KED encountered an I/O error while writing to the journal file. The error was not a file full error.

Your Response: Type Y and press RETURN to exit from the work session and guarantee preservation of your text file. Type N and press RETURN to continue the work session without the protection of a journal file.

In either case, examine the condition of the journal output device. For example, perform a DIRECTORY/BADBLOCKS operation to examine it for bad blocks.

Journal output file full—Exit session (Y/N)?

Explanation: The allocated space for the journal file is full.

Your Response: Type Y and press RETURN to exit from the work session and guarantee preservation of the text file. Type N and press RETURN to continue the work session without the protection of a journal file.

Journal preliminary buffer overflow—Exit session (Y/N)?

Explanation: The temporary journal file buffer is full.

Your Response: Type Y and press RETURN to exit from your work session and guarantee preservation of your text file. Type N and press RETURN to continue your work session without journal file protection.

?KED–F–Bad call to .VVV.V

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–File name conflict

Explanation: You entered a command line in which the journal file and output file have the same file specification.

Your Response: Rename either the journal file or the output file (typically, the journal file).

?KED–F–Insufficient memory

Explanation: KED:

KED cannot run, because too little memory is available (approximately 16K words are required) or KED may have malfunctioned.

KEX:

There is not enough available memory. Or you applied a customization (described in the *RT–11 Installation Guide*) to limit the amount of memory KEX requests with a .SETTOP, but the amount of memory you specified was too small.

Your Response: KED:

Switch to a system that provides more memory or switch to a smaller RT–11 monitor. In the *RT–11 System Message Manual*, the section entitled Increasing Storage and Memory Resources offers other suggestions that may be useful. If the message continues to appear, refer to Section F.2.

KEX:

Reapply the KEX customization, specifying a larger value for the highest address KEX requests with a .SETTOP.

?KED–F–Invalid device

Explanation: Your input or output device is not random access. The message does not apply to auxiliary files, which can reside on sequential-access devices.

Your Response: Use a random-access device for your input and output files.

?KED–F–I/O or device error—Continue (Y,N)?

Explanation: One of the following situations can cause that message.

Your device was switched off line.
Your output device was write locked.
Your volume has a bad block.

Your Response: Before responding, check that your device is on line. If it is an output device, check that it is write enabled.

To try the same function or command again, type Y and press RETURN. If the function or command works, continue editing. If the function or command does not work or if you type N and press RETURN, KED quits without saving your work and returns to the system prompt.

?KED–F–Journal not allowed with inspect mode

Explanation: You have requested a journal file when opening a file for inspection, which is not allowed.

Your Response: Reenter the command without requesting a journal file.

?KED–F–Output file shorter than input file

Explanation: The space available for your output file is smaller than your input file, or there may not be enough space on your volume.

Your Response: Use the SQUEEZE command to consolidate free space on your output volume or use a different volume.

?KED–F–RDBKW—No space to read into

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–RDFWD—No space to read

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–RDFWD—Pointer corrupt

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–SETUP—Allocation error

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–SPCFRE—Logic error, character count wrong

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–Temporary file shorter than input file

Explanation: The number of blocks you specified with a /BL option is less than the size of your input file.

Your Response: Reenter your command with a temporary file size that is greater than the input file size.

?KED–F–Too big to edit

Explanation: Your input file is too large. KED cannot edit or inspect a file that is larger than 32767 blocks. That restriction does not apply to auxiliary files or devices.

Your Response: Split your file into two or more parts, using auxiliary output files, and edit each part separately.

?KED–F–TTY??? Logic error

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–Unable to access input device

Explanation: KED did not access your auxiliary device for one of the following reasons.

Your device is invalid, not ready, or not installed.
The volume on your device has too many bad blocks.
There is no room for the device handler.

Your Response: Check your device abbreviation. If it is correct, check that the device is on line (use the SHOW/DEVICES command). If necessary, use the INSTALL command to install the device. If the message appears again, use the DIRECTORY/BADBLOCKS command to check the volume for bad blocks. If there are too many bad blocks, use another volume. If you cannot install the handler for the device, try the suggestions listed for ?KED–F–Insufficient memory.

?KED–F–Unable to access output device

Explanation: KED did not load the output device handler.

Your Response: If your device name is correct, check that your system volume contains the device handler. If the message appears again, try the suggestions listed for ?KED–F–Unable to access input device.

?KED–F–Unable to edit input file

Explanation: Your input file is too long. KED cannot edit a file that is larger than 16383 blocks.

Your Response: Split the file into two or more parts, using auxiliary output files, and edit each part separately.

?KED–F–Unable to open input file

Explanation: KED did not open your input file for one of the following reasons.

Your volume does not contain that file.
Your volume was not initialized.

Your input device is not on line.

Your Response: Check your file specification. If it is correct, check that the correct volume is installed and that the input device is on line.

?KED–F–Unable to open output file

Explanation: KED did not open your output file for one of the following reasons.

Your device is not ready or is write locked.
Your volume was not initialized.
Your volume does not have enough free space for the output file.
Your output file is protected.

Your Response: Make sure your output volume is properly installed and initialized. If that volume is too full, use another volume. To edit a protected file, use a different output file specification.

?KED–F–Unable to rename recover file to JBK file type

Explanation: You specified a journal file and a recover file of the same name. KED attempted to rename the recover file to type JBK, but failed, because a protected file of that name and type exists.

Your Response: Determine if the existing protected backup (JBK type) recover file is needed. If it is needed, unprotect it and rename it to a different file type (for example, OLD). Retry the operation.

?KED–F–Unable to trim output file

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRBKW—Bad block number

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRBKW—Block overrun

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRBKW—Plygnd not empty

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRFWD—Bad block number

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRFWD—Block overrun

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–WRFWD—Not enough data

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

?KED–F–Wrong version of RT–11

Explanation: You attempted to run a version (release) of KED that is not the same as the version (release) of RT–11 on your system.

Your Response: Replace the version of KED you are running with the version supplied on your current RT–11 software distribution.

?KED–W–Command file not found—Continue (Y,N)?

Explanation: KED did not find the command file on the volume you specified.

Your Response: Make sure you typed the input volume name and command file name correctly. To continue your work session, type Y and press RETURN.

?KED–W–File not found—Create it (Y,N)?

Explanation: KED did not find your input file on the volume you specified.

Your Response: To continue your work session, type Y and press RETURN. KED will create the file you specified on the volume you specified.

To end your work session, type N and press RETURN. Then correctly type the input volume name and file name.

?KED–W–Journal file exists—Replace (Y/N)?

Explanation: On the command line, you have requested the opening of a journal file that already exists.

Your Response: Type N and press RETURN to quit the work session; no output file or journal file is created, thereby preserving the existing journal file. Type Y and press RETURN to create a new journal file by replacing the existing file of the same name.

?KED–W–Journal file size below minimum—Continue (Y/N)?

Explanation: Because of a lack of free space on the journal file output volume, the journal file is smaller than the default (50_{10} blocks) or set size.

Your Response: Type Y and press RETURN to continue the work session with a smaller journal file size; the journal file may fill quickly. Type N and press RETURN to stop the work session. Perform a SQUEEZE operation on the journal file output volume or assign the journal file to a larger output volume and restart the work session.

?KED–W–Only MMMMMM blocks available for insertions—Continue (Y,N)?

Explanation: Free space on your output volume is less than 10 blocks larger than your input file. The message reports the number of blocks available.

Your Response: To continue your work session, type Y and press RETURN. KED will signal you when your input file becomes as large as possible. To cancel your work session and return to the system prompt, type N and press RETURN. Then use the SQUEEZE command to consolidate free space on your output volume, or specify an output volume that has more free space.

?KED–W–Only nnnnnn blocks available for journal file—Continue (Y,N)?

Explanation: Free space on your output volume for creating the journal file is less than 10 blocks.

Your Response: Type N or Y and press RETURN to cancel journaling.

If you type Y, you continue the work session without journal file protection.

If you type N, you are then asked if you want to continue the work session. Then type Y and press RETURN if you wish to continue the work session without journal file protection. Type N and press RETURN if you wish to cancel the work session. Then use the SQUEEZE command to consolidate free space on your output volume, or specify an output volume that has more free space.

?KED–W–Output file exists—Continue (Y,N)?

Explanation: Your output file name is the same as the name for a file on your output volume.

Your Response: To preserve the existing file and return to the system or KED prompt, type N and press RETURN.

To delete the existing file and create a new file with the same name, type Y and press RETURN.

?KED–W–Output file or device protected—Inspect (Y,N)?

Explanation: KED discovered one of the following.

The output file you specified for creating or editing exists as a protected file.
The input file you specified for editing is protected.
Your output device is write protected.

Your Response: To continue by inspecting the file you specified, on the device you specified, type Y and press RETURN.

To end your work session, type N and press RETURN.

?KED–W–Output files purged

Explanation: You used the QUIT command. If an auxiliary output file was open, KED does not save it. If you were editing or creating a file, KED does not save any of your work.

Your Response: Use any valid command or function.

?KED–W–Recover file not found—Continue (Y/N)?

Explanation: You specified a journal recovery operation and KED did not find the recover file on the specified (or default) volume.

Your Response: Type N and press RETURN to return to the monitor or KED utility prompt. Determine where the required recover file resides and retry the operation. Type Y and press RETURN to continue the work session without performing the journal file recovery operation.

If you continue the work session and exit from the text file, creating a new output file, you cannot subsequently perform a journal file recovery operation on that file.

?KED–W–Unable to open journal file—Continue (Y/N)?

Explanation: KED was unable to open the journal file on the default or specified volume.

Your Response: Check that the journal file output volume is available and on line. If running KEX as a system job, check that the volume's device handler is loaded. Retry the operation.

?KED–W–Unable to rename input file to BAK file type

Explanation: Your input file was not saved with the BAK file type, but KED attempted to save your output file.

Your Response: If you do not have another copy of your input file, you cannot recover it. You must use your output file if it exists.

?KED–W–Unable to set terminal characteristics—Continue (Y,N)?

Explanation: You are running KED on a multiterminal system, and your terminal was attached (.MTATCHed) by another job. KED now requires that you enter the command SET TERM NOCRLF.

Your Response: To continue your work session, type Y and press RETURN. To end your work session, type N and press RETURN. When KED returns to your system prompt, enter the command SET TERM NOCRLF. Then begin another work session.

LOAD command is invalid in LEARN mode

Explanation: You attempted to define two macros at the same time.

Your Response: Finish the LEARN operation before loading another macro

Learn buffer full

Explanation: The functions and commands you typed since you entered the LEARN command have too many bytes (characters) for the macro buffer. KED stops storing your macro in that buffer.

Your Response: Do not use unnecessary ADVANCE or BACKUP functions. Use abbreviations for commands, and use the shortest possible search strings. Other suggestions appear in Chapter 6. Appendix C shows the minimum number of bytes (characters) required for each function and command.

Learn macro already exists

Explanation: You attempted to assign an identifier to a learn buffer that is already assigned.

Your Response: Choose another identifier or clear the existing learn buffer and reuse it.

Local symbol definition too long

Explanation: In processing your LOCAL command, KED detected a local symbol that is more than six digits long.

Your Response: Check that the local symbol is correct.

Logic error in ADJUST

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in CHNGCASE

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in CUT

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in FILL

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in PASTE

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in SUBSTITUTE

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in undelete

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in undelete setup

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Logic error in WRITE SELECT

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Model:

Explanation: You pressed GOLD and FIND.

Your Response: Type the character or character string you want KED to search for, and then press ADVANCE or BACKUP. Your search string can be up to 60 characters long.

Model prompt canceled

Explanation: You pressed CTRL/Z or CTRL/C to cancel a model (search target) and the model prompt.

Your Response: Use any valid function or command.

Move failure in LOCAL

Explanation: You stopped the processing of your LOCAL command by pressing CTRL/C, or KED did not finish processing that command for some other reason.

Your Response: If you stopped the LOCAL command processing, you can use any valid function or command. If KED did not finish the processing for some reason, end your work session, begin it again, and try another LOCAL command. Correct any erroneous local symbols in the current LSB. Delete any ^A and ^D markers left by the LOCAL process.

Move PAGE or SECTION finds extremity of file

Explanation: A PAGE or SECTION function moved the cursor to the top or bottom of your file.

Your Response: Use any valid function or command.

Move to bottom when at bottom

Explanation: The BOTTOM function is not valid when the cursor is at the bottom of your file.

Your Response: Use any valid function or command.

Move to top when at top

Explanation: The TOP function is not valid when the cursor is at the top of your file.

Your Response: Use any valid function or command.

Nesting error

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

No journal file open

Explanation: You entered a command that applies to journaling, such as SET FREQUENCY, and journaling is not being used with this work session.

Your Response: If you thought you were keeping a journal file, exit from your work session and restart the session with journaling.

No macro by that name to execute

Explanation: You did not use a LEARN command to define the specified macro, or the specified macro failed while you were defining it.

Your Response: Use the LEARN command to define the specified macro.

No macro by that name to save

Explanation: You did not use a LEARN command to define the specified macro, or the specified macro failed while you were defining it. In either case, the specified macro does not exist to be saved.

Your response: Use the LEARN command to define the specified macro and retry the save operation.

No macro to execute

Explanation: The macro buffer is empty. You did not use a LEARN command to store your macro, or the last macro you tried to store failed while you were defining it.

Your Response: Use the LEARN command to store the macro you want to use.

No model defined

Explanation: KED did not process a FINDNEXT, REPLACE, or SUBSTITUTE function, because you did not specify a search string.

Your Response: To search for a character string, press GOLD and FIND, type the string, and then press ADVANCE or BACKUP.

You can also use any other valid function or command.

No room for this level of nesting

Explanation: You have attempted to nest more than three macros and KED allows a macro nesting depth of three.

Your Response: Do not nest macros beyond the third level.

No select range defined

Explanation: You used a REPLACE, APPEND, or CUT function without selecting text, and the cursor was not located at a search target.

Your Response: Use any valid function or command.

No select range defined for FILL

Explanation: You did not select text or you were not selecting text before you used the FILL function or command.

Your Response: Select the text you want to reformat before using the FILL command or function.

No select range for ADJUST

Explanation: You did not select text, or there are no characters in the part you selected before using the TABS ADJUST command.

Your Response: Select the text you want to adjust before using the TABS ADJUST command.

No select range for WRITE SELECT

Explanation: You used the SELECT function and then entered the WRITE SELECT command, but there are no characters in the part you selected.

Your Response: Select all the text you want to copy to your auxiliary output file, and then enter the WRITE SELECT command.

Not enough file space to do PASTE

Explanation: KED did not complete a PASTE function, because your output file is not large enough for the text in the paste buffer. The cursor is to the right of the last character KED copied from that buffer.

Your Response: See Section F.3.

You can also use the EXIT or QUIT command or any functions or commands that delete text from your file.

Not enough space to undelete

Explanation: KED did not complete an UNDELLINE, UNDELWORD, or UNDELCHAR function, because your output file is not large enough for the additional text. The cursor is to the right of the last character KED copied.

Your Response: See Section F.3.

Nothing to undelete

Explanation: KED did not process an UNDELLINE, UNDELWORD, or UNDELCHAR function, because the line, word or character buffer is empty.

Your Response: Use any valid function or command.

OPENLINE finds file full

Explanation: KED did not process an OPENLINE function, because your output file is not large enough for a carriage return and line feed.

Your Response: See Section F.3.

Operation is invalid during inspect

Explanation: While you are inspecting a file, you attempted an operation or command that would modify the file. KED cannot complete functions or commands that modify the file.

Your Response: Use another function or command to move the cursor or end your inspection.

Repeat:

Explanation: You pressed GOLD and typed a number on your keyboard. It indicates you are using the repeat function to repeat an editing operation or the special insertion function to insert a character you cannot insert directly.

Your Response: To complete the special insertion function, press GOLD and SPECINS. To complete the repeat function, press one or more of the other keypad keys (including GOLD, but not <x).

Saveset error

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Search canceled

Explanation: You pressed CTRL/C or CTRL/Z to cancel the FIND function and `Model:` prompt.

Your Response: Use any valid function or command.

Select range too large to CUT

Explanation: KED did not complete a CUT or APPEND function, because the paste buffer is not large enough for all the text you want to store.

Your Response: Select smaller sections of text, or use auxiliary output and input files for the same purpose as the CUT, APPEND, and PASTE functions.

You can also use any valid function or command.

TAB indent value wrong for align

Explanation: You tried to use the A function, but the column number of the cursor's position is not evenly divisible by your indentation setting.

Your Response: Check that the cursor's position and your indentation setting are correct. Then, use the A function again.

Tabs not enabled for ADJUST

Explanation: You used the TABS ADJUST command before using the SET TABS command.

Your Response: Enable structured tabs with the SET TABS command, and then use the TABS ADJUST command.

Target not found

Explanation: A FIND or FINDNEXT function moved the cursor to the top or bottom of your file without finding a character string that matches the string you specified.

Your Response: Use any valid function or command.

Too many arguments for command

Explanation: KED did not recognize your command, because you added extra characters after the last valid word or argument in the command.

Your Response: Use any valid function or command.

Unable to access auxiliary device

Explanation: KED did not access your auxiliary device for one of the following reasons.

Your device is invalid, not ready, or not installed.
The volume on your device has too many bad blocks.
There is no room for the device handler.

Your Response: Check your device abbreviation. If it is correct, check that the device is on line (use the SHOW/DEVICES command). If necessary, use the INSTALL command to install the device you need. If the message continues to appear, use the DIRECTORY/BADBLOCKS command to check the volume for bad blocks. If there are too many bad blocks, use another volume. If you cannot install the handler for the device, try the suggestions listed for ?KED–F–Insufficient memory.

Unable to close auxiliary file

Explanation: KED did not close your auxiliary output file. Your output device may have caused the problem, or KED may have malfunctioned.

Your Response: Check that your output device is on line. You can also try the suggestions listed for Unable to access auxiliary device (above).

If the message continues to appear, see Section F.2.

Unable to delete mark

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Unable to delete target

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Unable to insert tab

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Unable to open auxiliary input file

Explanation: KED did not open your auxiliary input file for one of the following reasons.

Your volume does not contain that file.
Your volume was not initialized.
Your device is not ready.

Your Response: Check your file specification. If it is correct, check that the correct volume is installed and that your auxiliary input device is on line.

Unable to open auxiliary output file

Explanation: KED did not open your auxiliary output file for one of the following reasons.

Your volume was not initialized.
Your device is not ready.
Your device is write locked.

Your Response: Check that the auxiliary output volume is initialized. Also check that the auxiliary output device is on line and write-enabled.

Unable to replace symbol

Explanation: KED may have malfunctioned.

Your Response: See Section F.2.

Undelete buffer full

Explanation: KED did not complete a DELLINE or DELWORD function because:

You tried to delete more than the capacity of the line buffer, 132 characters.
or
You tried to delete more than the capacity of the word buffer, 80 characters.

Your Response: Use repeated DELLINE or DELWORD functions (and ignore the error signals), or use the paste buffer or auxiliary files.

You can also use any valid function or command.

Use the SET HELP command to enable help screens

Explanation: You pressed the help key (PF2) and the help screen is disabled from issuing the SET NOHELP command.

Your Response: If you wish to view the help screens, first issue the SET HELP command and then press PF2.

Valid start of LSB not found

Explanation: No global symbol or .ENABL LSB directive exists between the cursor's position and the preceding form-feed character or the top of your file.

Your Response: Make sure the cursor is in the first page of an LSB before you use the LOCAL command.

WORKING...

Explanation: When an operation requires more than the usual amount of time, KED flashes that message at the top of your terminal screen until the operation is completed. The length of your file, the device you are using, and the function you are using determine when that message appears.

Your Response: Wait until the operation is completed, unless you want to stop it by pressing CTRL/C twice.

WRITE finds end of file

Explanation: Your WRITE command copied the last character in your input file to the auxiliary output file.

Your Response: Use any valid command or function.

Index