



Brady



# THE UNIX SCREEN EDITOR

August Hansen

TP31

H249

8763914

# **vi—The UNIX Screen Editor A User's Guide**

***August Hansen***

**A Brady Book**  
Published by Prentice Hall Press  
New York, NY 10023

0100000

Copyright © 1986 by Brady Communications Company, Inc.  
All rights reserved  
including the right of reproduction  
in whole or in part in any form.

A Brady Book  
Published by Prentice Hall Press  
A Division of Simon & Schuster, Inc.  
Gulf + Western Building  
One Gulf + Western Plaza  
New York, New York 10023

PRENTICE HALL PRESS is a trademark of Simon & Schuster, Inc.

Manufactured in the United States of America

1 2 3 4 5 6 7 8 9 10

### **Library of Congress Cataloging-in-Publication Data**

Hansen, August.  
vi—the UNIX screen editor.

"A Brady book."

Includes index.

1. Text editors (Computer programs) 2. vi (Computer  
program) 3. UNIX (Computer operating system) I. Title.

II. Title: vi—the U.N.I.X. screen editor.

QA76.76.T49H36 1986 652'.5 86-9519

ISBN 0-89303-928-4



# **vi—The UNIX Screen Editor A User's Guide**

E8763914



8763914

## About the Author

August (Augie) Hansen is a consultant in the allied fields of computers and communications. He holds a Bachelor's degree in electrical and electronics engineering and has 20 years of experience in areas of technology involving these fields.

The majority of his work experience has been on voice and data switching systems and general purpose timesharing computer systems while working at AT&T's Bell Laboratories and Information Systems units. He has been involved in the design and development of computer and communications products and in the preparation of user documentation for them.

He is a contributing editor to *PC Tech Journal* and writes frequently on UNIX and communications topics related to the IBM PC family of microcomputers.

## Acknowledgments

The text and examples used in this book were prepared on several minicomputer and microcomputer versions of the UNIX system. The primary system was an AT&T PC6300 running SCO XENIX Version 3.0, Release 1.4. Editor versions ranging from 2.7 to 3.9 were used in different computer environments to test the examples.

I thank all of my former associates at Bell Laboratories and AT&T Information Systems for their support and for seven years of shared experiences using the UNIX system in its evolving forms in a production environment. I am particularly indebted to Carl Brandauer, Dave Custer, and Al Larson for their continuing inspiration and friendship, and to several anonymous reviewers for their insightful comments and suggestions.

I also wish to thank Deborah Corson for her hard work in pulling all the pieces together — a job that is unseen and unrecognized (unless it is done poorly). Thanks for doing it well.

And to my family, thanks for putting up with me during the struggle to meet my long-term objectives while trying to survive in the short term.

# Preface

This book is a distillation of my experiences in learning to use the visual editor when it first became available as an unsupported program within Bell Laboratories and in teaching others to use `vi` for programming and document preparation activities in a hectic production environment. Although `vi` is not necessarily the best tool for the job, it is the most widely available visual editor for UNIX(tm), and with AT&T's inclusion of `vi` in System V, Release 2, it is now officially supported.

Augie Hansen  
Denver, Colorado



# Introduction

Before we begin using `vi`, we will review a bit of the history behind it to put its use in perspective. We will also establish some useful conventions to be used throughout this guide.

---

## ***UNIX Editing***

---

The `vi` (usually pronounced VEE-EYE, although some users call it VIE) editor suits the needs of many computer users. Historically, programmers and software engineers have formed one of the largest populations of `vi` users. Writers and editors and word-processing specialists working on documents of all descriptions are a comparably sized group. In more recent times, managers, administrators, and general office workers have swelled the ranks of `vi` users under the UNIX operating system.

Early text editors for creating and modifying files were line oriented. Such editors, typified by `ed` on UNIX systems, have what seem like rather spartan capabilities compared to `vi`, but they are usually very efficient in their use of computing resources, and they are typically more powerful than their outward appearances suggest. It is unfortunate that they are far less efficient when measured by their contribution to user productivity. This is the legacy of programs that were designed to work at least acceptably with slow, dumb printing terminals.

---

## ***The Genesis of the `vi` editor***

---

The `ex` editor, an extended and enhanced version of the UNIX line-by-line editor, `ed`, was developed at the University of California at Berkeley by William Joy. Among its many features is the visual mode, which can be accessed from within `ex` or directly from the UNIX command line by typing `vi`. Beginning with the release of

UNIX System V, AT&T began to support `vi` as part of the product, although it had been used within AT&T and Bell Laboratories for about five years before being officially adopted.

`vi` has become available on many UNIX systems and has steadily gained popularity over the years in spite of challenges from full-screen editors of more recent vintage. Other screen editors for UNIX include EMACS in several varieties (all based on the editor used for the MIT MACS project); `se`, an AT&T word-processing style editor that is available on some UNIX systems (it's no longer supported), and an alphabet soup list of editors (`ned`, `red`, `scred`, `abe`, and so on) that have met with varying degrees of success. As the author of `scred` noted, it's very hard to come up with good names for editors because most of the obvious and fitting ones have been claimed.

In addition, PC/IX and IX/370, licensed versions of the UNIX system for the IBM PC and IBM 370, use `INed`, a full-screen editor with some interesting features and a combination of command and menu user interfaces. Emerging Technology Consultants produces the Professional Writers' Package, which includes its EDIX screen editor and WORDIX, a text processor, that has gained a strong following in the PC-DOS and MS-DOS operating system world. The product family is now sold by AT&T for use with its 3B line of UNIX systems and is sold on an OEM basis for other UNIX systems. Details about several of the most widely used of these editors can be found in Appendix D.

## — *Variations in vi Implementations* —

Numerous versions of `vi` are currently in use in the field. Where differences exist in the availability of commands or in the behavior of them, they will be noted in the text.

Because the `vi` program is provided in source form to suitably licensed customer sites, there may be other variations that were produced to fit `vi` onto a given machine or to provide modified or additional features. Your system administrator or a resident expert (often called a UNIX “guru”) should be able to help you find out what local customs and features are in effect. Don't be timid about asking for help if you need it.



If you are the system administrator, read about and practice using `vi` as much as possible. With only a modest effort, you will begin to feel comfortable with `vi`. You'll find that helping others with their editing problems will force you to learn how to use it extensively and efficiently.

---

## ***Host Environments***

---

An advantage of learning to use `vi` is that the popularity and availability of UNIX is steadily increasing, and the `vi` editor is supported on the majority of UNIX installations. The effort you put into learning to use `vi` and related UNIX tools will most likely not be wasted when you move to a different UNIX system. Because `vi` is the screen editor officially supported by AT&T, it will likely be available on most UNIX systems you will use.

The numbers and types of microcomputers running UNIX have grown very quickly in the last few years. A representative sampling of microsystems includes the TRS-80 Model 16, most of the IBM PC family and compatible machines, intelligent work-stations from Sun Microsystems, a range of micro-based computers from AT&T, Hewlett-Packard, and even the Data General/One lap portable.

In addition, the range of minicomputers running UNIX is astounding. There are UNIX versions for the DEC PDP-11 and VAX-11 series, some Data General offerings, machines provided by Perkin-Elmer, Gould, Convergent Technology, and many other companies.

UNIX even runs on some mainframe computers, including an Amdahl (running UTS), the IBM 370 family running Maxi-UNIX and IX/370, and some of the Cray supercomputers.

---

## ***Text and Keyboard Notation***

---

Throughout this guide, we will use special typefaces and symbols to identify user input, system responses, and special characters. Certain control and special characters are understood by `vi`. A

control character is a two-key combination requiring you to press and hold the key labeled CONTROL (or CTL, CTRL, CNTRL, and such), then press another key briefly before releasing both keys. The CONTROL key sequence will be designated by the symbol `^` followed immediately by a single character, displayed as an uppercase letter, although you may type the letter in either case. For example, the command to redraw the screen is `^L`, which means to hold the CONTROL key and press the “L” key, then release both keys. Use of the SHIFT key or CAPSLOCK mode do not alter the meaning of the key sequence. This confusing way of symbolizing control characters is due to tradition born in the days of uppercase-only terminals.

Because the symbol for a control character usurps the caret symbol (`^`), in this book I will use `↑` to designate a caret and reserve the `^` symbol for use in indicating control characters only. This style is consistent with the prevailing practice in most UNIX documentation and its application programs.

Your input, that which you type on the keyboard, is shown in bold type. Components of your input that are variable, such as text or file names, are shown in bold italic type. For example, the command to edit a file will be shown as

***vi filename***<CR>

which has ***vi*** in bold meaning literal input and ***filename*** in bold italics indicating that you will supply the name of the file you want to edit.

Commands to the UNIX shell and some within ***vi*** are terminated with a carriage return. The return key is marked in various ways on different terminals and computers, so I will use the the sequence <CR> to symbolize whatever key on your keyboard produces the carriage return. Some terminal and computer keyboards have keys marked ENTER, LINEFEED, or a down arrow or left arrow. You may have to experiment a bit on your equipment to find out which key to use to terminate a command.

Similarly, I will use <BS> to symbolize a backspace key, which may be marked with the letters BS, BACK, or a left-facing arrow on some keyboards, and may not be present on others. The control-key sequence `^H` produces the same internal code as a backspace key.

Editor responses will usually be shown as normal text except when the response would be displayed in some highlighting mode, which will be simulated in this text by bold printing with a continuous underline (**like this**).

Describing an editor as complex as `ex/vi` poses a chicken and egg problem. It is impossible to avoid mentioning some topics in limited, specific uses before they are explained in detail in some later section. Forward references are used in such cases. Also, some detailed material is only alluded to in the main text. Details for those who care to read them have been placed in the appendices. After you have become proficient with the basics of editing, you should take a pass through some of the hidden detail to find things that may help you in your editing tasks.

With the formalities out of the way, it's time to begin editing with `vi`. The lessons assume that you have no specific knowledge of any editor. Each lesson identifies learning goals and has a set of helpful exercises at the end to test your knowledge of the material. To get the most benefit from the exercises, try to do them, preferably on a functioning system, to get a feel for how well you understand the lesson material.

## **Registered Trademarks**

UNIX is a trademark of AT&T Bell Laboratories.

XENIX and Microsoft are trademarks of Microsoft Corporation.

DataGeneral/ONE is a trademark of Data General.

WordStar is a trademark of Micropro International Corporation.

Venix is a trademark of VenturCom, Inc.

IBM is a trademark of International Business Corporation.

EDIX, WORDIX, and Professional Writer's Package are trademarks of Emerging Technology Consultants, Inc.

DEC, VAX, and PDP are trademarks of Digital Equipment Corporation.

Multics is a trademark of Honeywell.

## **Limits of Liability and Disclaimer of Warranty**

The author and publisher of this book have used their best efforts in preparing this book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

# Contents

## **Preface xi**

## **Introduction xiii**

- UNIX Editing xiii
- The Genesis of vi xiii
- Variations in vi Implementations xiv
- Host Environments xv
- Text and Keyboard Notation xv

## **Lesson 1. Getting Started 1**

- 1.1 Initial Setup 1
- 1.2 Creating Some Text 2
- 1.3 Starting vi 5
- 1.4 Viewing the Text 7
- 1.5 Saving Changes 9
- 1.6 Quitting vi 11
- 1.7 Abandoning Changes 11



## **Lesson 2. Positioning the Cursor 13**

- 2.1 Character and Line Motions 14
- 2.2 Word Motions 16
- 2.3 Moving to In-Line Objects 17
- 2.4 Moving to General Text Objects 18
- 2.5 Moving to Window Positions 24
- 2.6 Marking and Returning to Buffer Positions 24

## **Lesson 3. Editor Modes and Window Control 27**

- 3.1 Editor Structure 27
- 3.2 vi Command Mode 30
- 3.3 vi Text Insert Mode 30
- 3.4 ex Modes 34
- 3.5 General vi Command Syntax 34
- 3.6 General ex Command Syntax 35



3.7 Window Control 37

## ***Lesson 4. Modifying Text 41***

4.1 Repeating Commands 41

4.2 Deleting Text 41

4.3 Changes, Replacements, and Substitutions 46

4.4 Copying Text 50

4.5 Moving Text 52

4.6 Using Named Buffers 52

## ***Lesson 5. Using Insert-Mode Commands 57***

5.1 Error-Correcting Commands 57

5.2 Special Insert-Mode Commands 59

5.3 Quoting Special Characters 63

## ***Lesson 6. Recovering from Errors 67***

6.1 Undoing Commands 67

6.2 Retrieving Deleted Lines 68

6.3 Recovering Files After Interrupts and Crashes 69

6.4 Preserving a File 71

6.5 Refreshing the Screen 71

6.6 Recovering from Internal Editor Errors 73

## ***Lesson 7. Working with Files 75***

7.1 Writing to a File 75

7.2 Reading from a File or Command Output 76

7.3 Editing a File 81

7.4 Alternate File Editing 81

7.5 Transferring Text Between Buffers 82

7.6 Sequential Editing 85

## ***Lesson 8. Editor Variables, Command Lines, and Shells 89***

8.1 vi Editor Variables 89

8.2 UNIX Command-Line Options for vi 94

8.3 Types of Shell Escapes 95

8.4 Filtering the Buffer	97
<b>Lesson 9. Miscellaneous Commands</b>	<b>101</b>
9.1 Editor Statistics and File Information	101
9.2 Using Line Numbers	103
9.3 Case Conversions	104
9.4 Joining Lines	105
9.5 Transposing Characters	106
9.6 Shifting Objects	107
<b>Lesson 10. Using Other Editing Modes</b>	<b>111</b>
10.1 The <code>ex</code> Editor	111
10.2 Invoking the <code>ex</code> Editor	112
10.3 Open Editing Mode	114
10.4 Line-Oriented Editing with <code>ex</code>	116
<b>Lesson 11. Special Features</b>	<b>127</b>
11.1 Macros in Command Mode	127
11.2 Abbreviations and Mapped Strings	131
11.3 Tags and Tag Files	132
<b>Command Reference</b>	<b>137</b>
<b>Glossary</b>	<b>183</b>
<b>APPENDICES</b>	
A Supported Terminals	187
B Customizing the Editor	193
C Writing Termcap Entries	197
D Regular Expressions and Their Uses	201
E Other Screen Editors for UNIX and XENIX	205
F An On-Line <code>vi</code> Help System	213
<b>Index</b>	<b>225</b>

## LESSON

# 1

# Getting Started

### Goals

- Prepare your UNIX working environment for using the `vi` screen editor
- Create some text in a file for editing
- View text using scrolling and paging commands

## 1.1 Initial Setup

Before using `vi`, you must do a few things. `vi` has to know what type of terminal it is interacting with so it can correctly display characters on the screen and interpret input from the keyboard. Each terminal type has its own way of doing things. Programs like `vi` automatically modify the way they work to accommodate a wide range of terminals.

To find out whether a terminal type has been set in your UNIX working environment, type `echo $TERM<CR>`. If no value has been set, UNIX will reply with a new shell prompt on a new line. If the `TERM` variable has been assigned a value, it will be displayed. Be sure the value is an appropriate one for the terminal you are using. See Appendix A for a list of terminal types recognized by most computer systems running UNIX or a UNIX work-alike operating system.