

(影印版)

第二版  
上卷

# sendmail



REILLY®  
电力出版社

*Bryan Costales  
with Eric Allman*

---

sendmail  
*Second Edition*

上卷



O'REILLY®  
*Beijing · Cambridge · Farnham · Köln · Paris · Sebastopol · Taipei · Tokyo*

# 出版说明

因特网的迅猛发展和开源软件运动的兴起是对计算机工业界具有深远影响的革命性事件。事实上，众多因特网关键技术正是开源软件运动的累累硕果。

对于软件业还不发达的中国而言，开源软件运动无疑为我们提供了一次难得的机遇。而因特网的深入人心，也使我们充分地吸收整个人类社会的精神文明财富、缩小与世界先进水平的差距成为可能。

为了满足国内读者了解网络和开源软件技术的迫切需要，中国电力出版社在世界著名计算机图书出版机构 O'Reilly & Associates 公司的鼎力襄助下，影印出版了一批该公司久负盛名的英文经典技术专著。

由于社会制度和文化背景的差异，这些著作中可能会存在不甚符合我国实际情况的问题，希望读者注意分析辨别。

作为一项长期的出版工程，我们将始终关注全球计算机技术的最新发展，如果您对选题或书的内容有任何意见、建议，欢迎您与我们联系。

中国电力出版社计算机编辑室

Email: cepp01@mx.cei.gov.cn

2000 年 5 月

---

## *Preface*

The primary reason for this book, the second edition of *sendmail*, is the release of version 8.8 of the *sendmail* program. V8.8 *sendmail* differs so significantly from earlier versions that a complete rewrite was called for.

Also, since the publication of the first edition, feedback from around the world made it clear that a number of improvements could be made. Some readers felt, for example, that the original offered too little in the way of real-world help, and that it lacked an annotated, major configuration file. Other readers offered suggestions of ways to make the book clearer. Some of these were minor but others, like the treatment of options, required extensive rewriting. One of the most noticeable changes between this edition and the original is the addition of section numbers and their attendant cross referencing.

## *Preface*

King Gordias of Phrygia once created a knot so tangled that no one could undo it. The Gordian Knot stayed just that, or so the story says, until Alexander the Great came along and took a different approach to untying knots. It would be nice if the knot that is *sendmail* could be undone with one quick stroke of fresh insight, but, alas, it cannot. Instead, a more mundane approach must be taken, so in this book we untie it the hard way, one strand at a time.

But, you may ask, “Why the effort? Doesn’t *sendmail* predate the dawn of computing time? Isn’t it time to replace it with something new, something better, something modern?” Not so. Age has brought *sendmail* maturity and reliability. The *sendmail* program has withstood the test of time because it is more than just a program, it is a philosophy: a general-purpose, internetwork mail-routing facility

with the flexibility and configurability to solve the mail-routing needs of all sites large or small, complex or simple.

These strengths of *sendmail* are also its weaknesses. Configurability has bred complexity. The *sendmail* program is difficult to configure and even more difficult to understand. Its configuration file, for example, can be positively frightening. But don't despair. With this book in hand, you should be able to configure *sendmail* to meet any need and bring the days of the *sendmail* guru to an end.

## History

The *sendmail* program was originally written by Eric Allman while he was a student and staff member at the University of California at Berkeley. At the time, one campus machine (*Ingres*) was connected to the ARPAnet, and was home to the INGRES project where Eric was working. Another machine (*Ernie CoVax*) was home to the Berkeley UNIX project and had recently started using UUCP. These machines (as well as several others on campus) were connected by a low-cost network built by Eric Schmidt, called BerkNet. Software existed to move mail within ARPAnet, within UUCP, and within BerkNet, but none yet existed to move mail between these three networks.

A sudden increase in protocol types, coupled with the anticipation of an explosion in the number of networks, motivated Eric to write *delivermail*—the precursor to *sendmail*. The *delivermail* program was shipped in 1979 with 4.0 and 4.1 BSD UNIX. Unfortunately, *delivermail* was not flexible enough to handle the changes in mail routing requirements that actually occurred. Perhaps its greatest weakness was that its configuration was compiled-in.

In 1980, ARPAnet began converting from NCP (Network Control Protocol) to TCP (Transmission Control Protocol). This change increased the number of possible hosts from 256 to over one billion. Another change converted from a “flat” host-name space (like MIT-XX) into a hierarchical name space (like XX.MIT.EDU). Prior to these changes, mail was transported using the *ftp* protocol (File Transfer Protocol). Afterward, a new protocol was developed for transporting mail called SMTP (Simple Mail Transfer Protocol). These developments were not instantaneous. Some networks continued to run NCP years after most others switched to TCP. SMTP itself underwent many revisions before finally settling into its present form.

Responding to these and other changes, Eric evolved *delivermail* into *sendmail*. To ensure that messages transferred between networks would obey the conventions required by those networks, Eric took a “liberal” approach—modifying address information to conform, rather than rejecting it. At the time, for example, UUCP mail often had no headers at all, so *sendmail* had to create them from scratch.

The first *sendmail* program was shipped with 4.1c BSD (the first version of Berkeley UNIX to include TCP/IP). From that first release to the present,\* Eric has continued to enhance *sendmail*, first at UC Berkeley, then at Britton Lee, then back at UC Berkeley, and now with InReference Inc. The current version of *sendmail* is 8.x (or V8 for short). V8 is a major rewrite that includes many bug fixes and significant enhancements.

But Eric wasn't the only one working on *sendmail*. In 1987, Lennart Lovstrand of the University of Linköping, Sweden, developed the IDA enhancements to BSD *sendmail* Version 5. IDA (which stands for "Institutionen för Datavetenskap") injected a number of improvements into *sendmail* (such as support for *dbm* files and separate rewriting of headers and envelopes) and fixed a number of bugs. As the '90s approached, two offsprings of IDA appeared.

Neil Rickert (Northern Illinois University) and Paul Pomes (The University of Illinois) took over maintenance of IDA *sendmail*. With contributions from around the world, their version (UIUC IDA) represents a continuation of the work begun by Lennart Lovstrand. Neil focused on fixing and enhancing the configuration files into their current *m4*-based form. Paul maintained the code, continually adding enhancements and fixing bugs. In general, their version was large, ambitious, and highly portable. It succeeded in solving many complex mail routing problems.

A variation on IDA *sendmail* was also developed by Paul Vixie (while at Digital Equipment Corporation). Called KJS (for King James *sendmail*), it was a more conservative outgrowth of Lennart Lovstrand's last IDA release. The focus of KJS was on code improvement rather than changes to configuration files.

In addition to these major offshoots, many vendors have modified *sendmail* to suit their particular needs. Sun Microsystems made many modifications and enhancements to *sendmail*, including support for *nis* and *nisplus* maps. Hewlett Packard also contributed many fine enhancements including 8BITMIME support.

This explosion of *sendmail* versions has led to a great deal of confusion. Solutions to problems that work for one version of *sendmail* fail miserably with others. Beyond this, configuration files are not portable, and some features cannot be shared.

In 1994, Eric began work on V8.7 *sendmail*. The first major departure from tradition in years, V8.7 introduces multicharacter option and macro names, new interactive commands to use with *-bt* mode, and fixes many of the problems and limitations of earlier releases. But, more important, V8.7 has officially adopted most of the good features from IDA, KJS, Sun, and HP's *sendmail*, and kept abreast of the latest standards from the Internet Engineering Task Force. In 1996, Eric

---

\* With one long gap between 1982 and 1990.

began work on V8.8 *sendmail*. This release continued the trend begun with V8.7, adding many requested new features and options, and tightening security. Since V8.8 is now the official release of *sendmail*, it is the one solely documented in this book.

## *Eric Allman Speaks*

I have to admit that I'm surprised by how well *sendmail* has succeeded. It's not because of a large marketing organization or a deep-pockets budget. I think there are three reasons.

First, *sendmail* took the approach that it should try to accept, clean up, and deliver even very "crufty" messages instead of rejecting them because they didn't meet some protocol. I felt this was important because I was trying to gateway UUCP to the ARPAnet. At the time, the ARPAnet was small, UUCP was anarchy (some say it still is), and UNIX mail programs generally didn't even understand headers. It was harder to do, but after all, the goal was to communicate, not to be pedantic.

Second, I limited myself to the routing function—I wouldn't write user agents or delivery back-ends. This was a departure from the dominant thought of the time, where routing logic, local delivery, and often the network code was incorporated directly into the user agents. But it did let people incorporate their new networks quickly.

Third, the *sendmail* configuration file was flexible enough to adapt to a rapidly changing world: The 1980s saw the proliferation of new protocols, networks, and user agents.

And, of course, it didn't hurt that it was free, available at the right time, and did what needed to be done.

Configuring *sendmail* is complex because the world is complex. It is dynamic because the world is dynamic. Someday *sendmail*, like X11, will die—but I'm not holding my breath. In the meantime, perhaps this book will help.

When I started reviewing Bryan's manuscript, I had been avoiding any major work on *sendmail*. But then I started reading about various petty bugs and annoyances that all seemed easy to fix. So I started making small fixes, then larger ones; then I went through RFC1123 to bring the specs up-to-date, cleaned up a bunch of eight-bit problems, and added ESMTP. It would be fair to say that the book and *sendmail* Version 8 fed on each other—each has improved the presence of the other.

## *The Future*

The explosive growth of the Internet, coupled with the universal acceptance of email as a desirable form of communication, guarantee that *sendmail* will continue to evolve and improve for years to come. Well before the release of V8.8, Eric was already thinking about V8.9. To glimpse some of what is in store, look at the file `KNOWNBUGS` distributed with the source. Note that some of the problems listed are thorny in the extreme and difficult to solve.

Also, percolating constantly in the back of Eric's brain is something he calls "Son of Sendmail" (or *sos*), a complete rewrite of *sendmail* from the ground up targeted for the far future. Those of you who have followed the evolution of the source may have noticed that functions are constantly being redesigned to make a future, massive rewrite easier and more modular.

## *Organization*

We've divided this book into four parts, each addressing a particular aspect of *sendmail* as a whole.

### *Part I — A Tutorial*

Part One is a tutorial that can serve as either a hands-on, step-by-step introduction to *sendmail* for the beginner or a succinct review for the more experienced user. Chapters 1 through 4 form an overview of email in general, and discuss the roles, behavior, and parts of *sendmail*. Chapters 5 through 15 examine the configuration file in detail. In them, we develop a mini-configuration file, suitable for use on some client workstations. Chapters 16 and 17 conclude the tutorial, tying up loose ends and transitioning into more complex configuration files.

### *Part II — Installation*

Part Two covers compilation and installation of *sendmail*. Chapter 18 shows how to compile and install *sendmail* from the source. Chapter 19 shows how to create a configuration file with the *m4*(1) configuration technique. Chapter 20 concludes by illustrating the *checkcompat*() routine.

### *Part III — Administration*

Part Three covers general administration of *sendmail* for more experienced users. Chapter 21 covers DNS in general and MX records specifically. Chapter 22 shows how to protect your site from intrusion. Chapters 23 through 26 round out the picture with details about the queue, aliases, mailing lists, logging, and statistics.



## ***Part IV — Reference***

Part Four is the nitty-gritty—a reference section that provides more detail about *sendmail* than you may ever need. Each chapter is dedicated to a specific aspect of *sendmail* or its configuration file. Chapter 31, for example, details defined macros, and includes an alphabetized reference. Chapter 37 shows all the debugging switches.

## ***Audience and Assumptions***

This book is primarily intended for system administrators who have been granted the dubious honor of administering email. But not all UNIX systems are managed by administrators. Many are managed by programmers, network engineers, and even by inexperienced users. It is our hope that this book satisfies all of you, no matter what your level of experience.

The true beginner should begin with Part 1, skipping ahead as needed.

The beginning system administrator should probably start with the tutorial in Part 1, then read Part 3 for help in understanding how to administer *sendmail*. Note that Parts 2 and 4 will reveal answers to many nagging questions that seem to be otherwise unanswered.

The experienced system administrator who wants to install and manage V8 *sendmail* should read Parts 2 and 3 first to gain the needed background. Then read Part 4.

UNIX gurus and *sendmail* specialists should find Part 4 to be of value (even Eric keeps a copy on his desk). In it, every arcane detail of *sendmail* is listed alphabetically. There is, for example, a single chapter dedicated to options, with every option listed and explained.

No matter what your level of expertise, the sheer size of this book forces us to assume that you are familiar with the day-to-day system workings of UNIX. If not, you must learn UNIX elsewhere.

## ***UNIX and sendmail Versions***

For the most part, we illustrate *sendmail* under BSD UNIX and its variants (such as SunOS 4.x). Where AT&T System V (SysV) differs (such as Sun's Solaris 2.x), we illustrate those differences.

Our primary focus throughout this book is on version 8.8 *sendmail*. For completeness, and where necessary, we also discuss V8.7 and earlier (such as BSD's version 5,\* IDA, early Sun, Ultrix, and NeXT).

## Conventions Used in This Handbook

The following typographic conventions are used in this book:

**Italic** is used for names, including pathnames, filenames, program and command names, usernames, hostnames, machine names, and mailing-list names, as well as for mail addresses. It is also used to emphasize new terms and concepts when they are introduced.

**Constant Width** is used in examples to show the contents of files or the output from commands. This includes examples from the configuration file or other files such as message files, shell scripts, or C language program source. Constant-width text is quoted only when necessary to show enclosed space; for example, the five-character "From " header.

Single characters, symbolic expressions, and command-line switches are always shown in constant-width font. For instance, the `o` option illustrates a single character, the rule `$-` illustrates a symbolic expression, and `-d` illustrates a command-line switch.

**Constant Bold** is used in examples to show commands or some other text that is to be typed literally by the user. For example, the phrase **cat/etc/sendmail.pid** means the user should type "cat/etc/sendmail.pid" exactly as it appears in the text or example.

**Constant Italic** is used in examples to show variables for which a context-specific substitution should be or will be made. In the string *Snum*, for example, *num* will be a user-assigned integer. In the output `error: num`, for example, *num* will be a variable value printed by a program (usually *sendmail*).

---

\* The versions jump from 5 to 8 because the managers of the BSD 4.4 UNIX distribution wanted all software to be released as version 8. Prior to that decision, the new BSD *sendmail* was designated Version 6. V6 survived only the alpha and beta releases before being bumped to V8.

% indicates a user shell.

# indicates a *root* shell.

## *Additional Sources of Information*

The source for the *sendmail* program comes with two documents by Eric Allman that are required reading. *Sendmail—An Internetwork Mail Router* provides an overview of *sendmail*, including its underlying theory. *Sendmail Installation and Operations Guide* provides installation instructions and a succinct description of the configuration file. Many vendors also provide online manuals which may reveal vendor-specific customizations not documented in this book. Also, if you have the source, see the files *RELEASE\_NOTES*, *src/READ\_ME*, and *cf/README*.

### *The RFCs*

A complete understanding of *sendmail* is not possible without at least some exposure to Request for Comments (RFC) issued by the Internet Engineering Task Force (IETF) at the Network Information Center (NIC). These numbered documents define (among other things) the protocols and operational requirements of the Internet. RFCs are available via anonymous FTP. See the Bibliography for information about how to retrieve individual RFCs.

#### *RFC821*

When *sendmail* transports mail from one machine to another over a TCP/IP network, it does so using a protocol called the *Simple Mail Transfer Protocol*, or SMTP for short. SMTP is documented in RFC821.

#### *RFC822*

The division of mail messages into a header and a body, as well as the syntax and order of header lines are all defined in RFC822, titled *Standard for the Format of ARPA Internet Text Messages*. It also describes the syntax of addresses.

#### *RFC819*

RFC819 is entitled *Domain Naming Convention for Internet User Applications* and describes the hierarchical form of host naming used today. This document defines the form a hostname must take.

#### *RFC976*

RFC976 is entitled *UUCP Mail Interchange Format Standard* and describes the format of mail messages transported between machines using UUCP.

*RFC1123*

RFC1123 is an extension to RFC821 and RFC822. It makes several amendments to the original documents and cleans up some previously ambiguous information.

*RFC1521 and RFC1522*

RFC1521 introduces and describes the standards for Multipurpose Internet Mail Extensions (MIME). MIME provides ways to embed non-text data (such as images, sounds, and movies) inside ordinary email messages.

*RFC1651, RFC1652, and RFC1653*

RFC1651 describes a general extension mechanism for SMTP, called ESMTP. RFC1652 describes an extension for transport of 8-bit data, called the 8BIT-MIME extension. RFC1653 describes an extension for message size declaration (see §34.8.22).

*RFC1891, RFC1892, RFC1893, and RFC1894*

RFC1891, *SMTP Service Extension for Delivery Status Notifications* (DSN), describes the ESMTP RCPT command's NOTIFY, RET, and ORCPT extensions. It also describes the ESMTP MAIL command's ENVID extension. RFC1892, *The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages*, describes the Content-Type: header requirements for DSN. It also describes the order and requirements for *multipart/report* MIME parts of a mail status report's mail message. RFC1893, *Enhanced Mail System Status Codes*, describes the meaning of the DSN status number fields. RFC1894, *An Extensible Message Format for Delivery Status Notifications*, describes the machine readable part's header-style keywords, for the machine readable part of the DSN return message described in RFC1892.

## *Other Books, Other Problems*

Two topics that are only touched upon in this book are The Domain Name System (DNS) and TCP/IP network communications. At a typical site, a significant number of problems with mail turn out to be problems with one of these other areas, rather than with *sendmail*.

The Domain Name System is well documented in the book *DNS and BIND, Second Edition* by Paul Albitz and Cricket Liu (O'Reilly & Associates, Inc., 1997).

The protocols used to communicate over the Internet are well documented in the book *TCP/IP Network Administration* by Craig Hunt (O'Reilly & Associates, Inc., 1992).

Finally, many mail problems can only be solved by the system administrator. The *sendmail* program runs as *root* and can only be installed and managed by *root*.

The art of functioning effectively as *root* is superbly covered in the *UNIX System Administration Handbook* by Evi Nemeth, Garth Snyder, Scott Seebass, and Trent R. Hein (Prentice Hall, 2nd edition 1995).

## Acknowledgments

First and foremost, I must thank George Jansen, who literally spent months turning my first horrendous, stream of consciousness prose into a form suitable for publication. He is truly the unsung hero of this work, and an editor extraordinaire.

Jon Forrest and Evi Nemeth both beat the tutorial chapters to death. Their feedback was extremely valuable in helping to trim and focus those chapters into a more useful form. Never let it be said that too many cooks spoil the broth—in this case they helped flavor it toward perfection.

Cricket Liu kindly reviewed the DNS chapter, where he found a few errors that somehow slipped by all the others. Bruce Mah and Sean Brennan were guinea pigs for the first and second editions respectively. They set up and ran *sendmail* based on early drafts and thereby uncovered omissions and mistakes that required correction. Gavin Cameron bravely applied the *checkcompat()* examples to real-world situations, thereby helping to debug that code for me. And John Funk kindly allowed alpha versions of *sendmail* to run in production mode at Mercury Mail.

Needless to say, this book would not have been possible at all if Eric Allman had not written *sendmail* in the first place. Every draft has passed through his hands, and he has spent many hours ensuring technical correctness, providing valuable insight, and suggesting interesting solutions to *sendmail* problems.

Neil Rickert, too, saw every draft of the 1st edition. But, alas, circumstances beyond his control forced him to bow out from the 2nd edition.

For this 2nd edition, Cricket Liu kindly reviewed Chapter 21, *DNS and sendmail*, and found several errors that slipped by everyone else.

Thanks and praise must go to Tim O'Reilly for agreeing to do this book in the first place. His experience has shaped this book into the form it has today. He was aware of the “big picture” throughout and kept his fingers on the pulse of the reader. Without his advice, a book this complex and massive would have been impossible.

Additional thanks must go to Lenny Muellner for tuning *troff* macros to satisfy the needs of this somewhat unique manuscript, and to Edie Freedman for gracefully accepting my unhappiness with so many cover designs except the current one, which I consider perfect.

The production folks at O'Reilly & Associates did a yeoman's job of achieving an outstanding finished book. So a special thank you to John Files for managing the project, Barbara Willette for copy editing, Nancy Kotary for help with final production, Kismet McDonough-Chan for her help in each phase of the production, Chris Reilley for the figures, Mary Anne Weeks Mayo for helping with quality control, Curt Degenhart, Madeleine Newell, and Ellie Fountain Maden for making the edits, Seth Maislin for doing the index, and Danny Marcus for proofreading.

Finally, thanks to a list of folks, each of whom helped in small but notable ways: Keith Johnson; Paul Pomes; Frederick Avolio; John Halleck; John Beck; Brad Knowles; Andrew Chang; Shau-Ping Lo; Nelson Morgan; Julene and Jake Bair; William Carter; all the folks at the International Computer Science Institute; John Funk, Dan Casson, and others at Mercury Mail; to many of those who sent interesting questions to the *sendmail-questions@sendmail.org* mailing list; and to all the postings to the *comp.mail.sendmail* news group.

---

# Table of Contents

## 上卷

<i>Perface</i> .....	<i>xv</i>
<i>I: A Tutorial</i> .....	<i>1</i>
<i>1: Introduction</i> .....	<i>3</i>
1.1 MUA Versus MTA .....	4
1.2 Why Is sendmail So Complex? .....	4
1.3 Three Important Parts .....	5
1.4 Run sendmail by Hand .....	6
1.5 The Header .....	8
1.6 The Body .....	8
1.7 The Envelope .....	9
1.8 Things to Try .....	10
<i>2: Have a V8</i> .....	<i>13</i>
2.1 Get the Source .....	13
2.2 Read the Documents .....	14
2.3 What's What in src .....	15
2.4 Preliminaries .....	16
2.5 Build .....	19
2.6 Test It .....	20
2.7 Things to Try .....	21

<b>3:</b>	<b><i>The Roles of sendmail</i></b>	<b>23</b>
3.1	Role in the Filesystem	23
3.2	Role in Local Delivery	27
3.3	Role in Network Transport	29
3.4	Role as a Daemon	30
3.5	Things to Try	31
<b>4:</b>	<b><i>How to Run sendmail</i></b>	<b>33</b>
4.1	Become a Mode (-b)	34
4.2	Verbose (-v)	39
4.3	Debugging (-d)	43
4.4	Things to Try	44
<b>5:</b>	<b><i>The sendmail.cf File</i></b>	<b>47</b>
5.1	Overview	47
5.2	The Minimal File	49
5.3	A Quick Tour	50
5.4	Things to Try	55
<b>6:</b>	<b><i>The Mail Hub and Delivery Agents</i></b>	<b>57</b>
6.1	The client.cf File	59
6.2	Define a Mail Delivery Agent	59
6.3	The Local Delivery Agent	61
6.4	Add the Missing Parts to Mhub	66
6.5	Things to Try	67
<b>7:</b>	<b><i>Macros</i></b>	<b>69</b>
7.1	Overview	69
7.2	Defining Macros	71
7.3	Predefined Macros	72
7.4	Things to Try	75
<b>8:</b>	<b><i>Addresses and Rules</i></b>	<b>77</b>
8.1	A Fictional Network	77
8.2	Why Rules?	81
8.3	Rule Sets	83
8.4	Rules	85
8.5	The Workspace	86
8.6	The Flow of Addresses Through Rules	87



8.7	Wildcard Operators .....	89
8.8	Things to Try .....	94
<b>9:</b>	<b>Rule Set 0 .....</b>	<b>95</b>
9.1	Introducing Rule Set 0 .....	96
9.2	The RHS Triple .....	96
9.3	Testing Rule Set 0 .....	99
9.4	The error Delivery Agent .....	100
9.5	Things to Try .....	101
<b>10:</b>	<b>Rule Set 3 .....</b>	<b>103</b>
10.1	Why Preprocess? .....	104
10.2	Rule Set 3 .....	105
10.3	Missing Addresses .....	108
10.4	Nested Angle Brackets .....	109
10.5	Details of Rule Flow .....	111
10.6	Things to Try .....	113
<b>11:</b>	<b>Rule Sets 1 and S=.....</b>	<b>115</b>
11.1	Flow of the Sender's Address .....	115
11.2	Rule Set S= .....	116
11.3	All Mail from the Hub .....	117
11.4	Rule Set Hubset .....	117
11.5	Testing So Far .....	120
11.6	Handling user@thishost .....	121
11.7	Rule Set 1 .....	123
11.8	Things to Try .....	124
<b>12:</b>	<b>Class .....</b>	<b>127</b>
12.1	The Class Command .....	128
12.2	The File Form of Class .....	135
12.3	Things to Try .....	138
<b>13:</b>	<b>Setting Options .....</b>	<b>139</b>
13.1	Options: An Overview .....	139
13.2	Required Options .....	140
13.3	Testing the Options .....	146
13.4	Sending Mail .....	147
13.5	Things to Try .....	148