

## Sendmail Evolution: 8.10 and Beyond

Gregory Neil Shapiro  
gshapiro@sendmail.org

Eric Allman  
eric@sendmail.com

*Sendmail, Inc.*  
6603Shellmound  
Emeryville, California

Originally presented at the  
1999 USENIX Annual Technical Conference  
FREENIX Track  
Monterey, CA

### ABSTRACT

*Sendmail*<sup>TM</sup> has been the *de facto* mail transfer agent implementation since the dawn of the Internet. Today, *sendmail* development is still driven by a continually changing set of network requirements and user demands. Lately, two new driving forces have also contributed to *sendmail* development. First, as more open source mail transfer agents, such as *Exim* and *Postfix*, become available, a new friendly competition has developed in which the authors of the various MTAs share their ideas via open source and help to advance open standards as opposed to advancing their own particular implementation. Second, a new “hybrid” company, Sendmail, Inc., has been created to offer commercial versions of the open source software while continuing to fuel open source development.

This paper will briefly discuss the evolution of *sendmail*; the influences which drive *sendmail* development; and how the creation of Sendmail, Inc. has contributed to the open source version. The paper will also describe the new features appearing in the next “functionality release” of open source *sendmail*. In particular, changes in queueing and new protocol support are discussed. Finally, the authors will speculate on future directions for *sendmail*.

## 1. Introduction

The *sendmail* mail transfer agent (MTA) is used on most UNIX<sup>TM</sup> systems today. Recent changes have influenced *sendmail* development, notably the creation of a new “hybrid” company dedicated to supporting both the open source code as well as a commercial version.

Section 2 gives a brief history of *sendmail*. Section 3 describes the forces acting to influence changes in *sendmail*. Section 4 outlines Sendmail, Inc.’s effects on the open source. Section 5 discusses changes appearing in *sendmail 8.10*. Future directions that *sendmail* may take are laid out in section 6. Finally, a summary and concluding remarks are presented in section 7.

## 2. History

To understand the continuing evolution of *sendmail*, you must first look at its history. Like many successful open source projects, *sendmail* started as a “scratch your itch” solution to a problem.

### 2.1. In the Beginning...

*Sendmail* started out as *delivermail*, written by Eric Allman, then a graduate student and staff member at the University of California at Berkeley. *Delivermail* solved the problem of routing mail between three different networks running on the Berkeley campus at the time: the ARPAnet, UUCP, and BerkNet. The first public version was distributed in 1979 as part of the Fourth Berkeley Software Distribution (4BSD) and later as part of 4.1BSD [Allm85].

Although *delivermail* solved the immediate problem faced by Berkeley, it was not generic enough to solve the problems of other custom networks in operation. Since the instructions for talking among the networks were part of the C source code, it was not easy for sites to reconfigure *delivermail* for their specific needs. The configuration was also not flexible enough to handle complex mail environments.

At the same time the ARPAnet was transitioning to the new Internet protocol, TCP/IP. Part of the new protocol suite included extracting mail transmission out of the file transfer protocol (FTP) into its own protocol, the Simple Mail Transport Protocol (SMTP) [RFC821].

The user demand for a customizable program and the network requirements created by the new mail protocol led to the creation of *sendmail*, which first

shipped in 1983 with 4.1c BSD—one of the initial operating systems to support TCP/IP. *Sendmail* accomplished two important goals. First, it provided a reference implementation of the Network Working Group (later the Internet Engineering Task Force, or IETF) mail standard [Cost97]. Second, the configuration was read at run time to allow reconfiguration for different networks without recompilation. Because of the wide variety of networks supported, the configuration was designed to be friendly toward non-conforming addresses. Instead of rejecting messages that were not acceptable to the standard, it tried to repair them; this broad acceptance of inputs maximized interoperability with other networks available at the time, such as UUCP.

By late 1986, Allman’s involvement with *sendmail* had tapered off, and several other people picked up development. The most important version was *IDA sendmail* from Lennart Lövstrand of the University of Linköping in Sweden, with later maintenance by Neil Rickert of Northern Illinois University and Paul Pomes of the University of Illinois [Cost97]. The most important feature added by IDA was the concept of external databases in DBM format. Shortly thereafter, Paul Vixie, then at Digital Equipment, created *KJS (King James Sendmail)*, an attempt to unify the divergent versions, but this version was not widely adopted. *Sendmail* had effectively splintered.

### 2.2. Sendmail 8 Emerges

In late 1989, Allman returned to U.C. Berkeley, and not long thereafter was drawn back into *sendmail* development. By July of 1991, serious work on what would become *sendmail 8* had begun. Many ideas were taken from *IDA sendmail* and *KJS*, although most were generalized. For example, external databases were added, but in such a way that formats other than DBM were available. *Sendmail 8.1* was released with 4.4 BSD in mid-1993. *Sendmail 8* quickly became a unifying influence, as vendors converted from their hacked versions to the newer version. Some features from vendor versions were also included in the new release, for example, NIS support from Sun Microsystems. These additions are just one of many examples of the success of open source software: *sendmail 8* was fertilized with ideas from other open source and vendor versions.

Another important change that occurred concurrent with *sendmail 8* was that versions were controlled more carefully. The previous major release (*sendmail 5*) had no fewer than 143 “dot” releases (that is, 5.1 through 5.143), often more than one in a single

day. Some of those were intended for public consumption, some were test releases. With version 8, *sendmail* switched to a policy of clearly labeling test releases, producing production releases less often, and clearly identifying new functionality releases from bug-fix releases. This change in release frequency was essential to the wide acceptance of *sendmail* 8 by the community. The downside of this change is that people who like to be on the “bleeding edge” have to wait longer, and new features are not tested immediately. We view this loss of quick feedback as being an acceptable tradeoff.

An unfortunate effect of the success of *sendmail* 8 was that Allman quickly became overloaded with answering questions. This overload was the impetus behind the establishment of the Sendmail Consortium, a loosely-knit group of volunteers providing free support for *sendmail*. Gregory Shapiro was invited to join that group during the 8.8 cycle, and by 8.8.6 was doing a large part of development and most of the release engineering, although Allman continued to review and approve changes.

In 1997, Allman found that even with the help of an extremely capable volunteer staff, he was unable to keep up with the support load and continue to move *sendmail* forward. After exploring several other approaches for adding resources for *sendmail* development, he finally settled on founding a “hybrid” business model company to produce a commercial version of *sendmail* while continuing to support and extend the open source version. By using the “hybrid” approach, he was able to protect the interests of the open source community while creating a viable business model.

### 3. Driving Forces

As can be seen in the preceding section, *sendmail* has responded to both changing network requirements and user demands. In addition to these demands, new open source MTA alternatives help in driving *sendmail* forward.

#### 3.1. Network Requirements

The network requirements come both from the changing face of the Internet and from new Internet drafts and RFCs from the IETF. For example, up until version 8.9, *sendmail* allowed third party, promiscuous relaying by default. This willingness to relay had been an acceptable, even desirable, default for more than 15

years. Unfortunately, with the growth of spam on the Internet, this default is no longer acceptable.

The increasing use of email as a vector of viruses has heightened the need for MTAs to include content checking. An SMTP server running on a firewall must be prepared to vet the data it is handling. Because of this need, 8.9 included message header checking and 8.10 will include a mail filter API for more advanced header and body filtering.

Changes in Internet standards from the IETF also have a major impact on *sendmail*. During 1998, the IETF accepted 22 new RFCs that involved electronic mail in one form or another. At least one of these RFCs has a direct effect on MTAs such as *sendmail*. RFC 2476, *Message Submission*, specifies a separate protocol for initial insertion of a new message into the message delivery system using SMTP [RFC2476].

As of April 1999, three more MTA-related RFCs have already appeared. RFCs 2487 [RFC2487] and 2554 [RFC2554] provide encryption and authentication for an SMTP session. RFC 2505 [RFC2505] is a set of recommendations for features that MTAs should provide to combat spam. Additionally, the Detailed Revision/Update of Message Standards (drums) IETF Working Group is preparing to release the long-awaited update to RFCs 821 and 822, probably later in 1999 [SMTPUPD, MSGFMT]. Clearly, the messaging standards landscape is not static.

#### 3.2. User Demands

By far, however, most feature requests come from *sendmail* users. It is common for the Sendmail Consortium to receive three to five feature requests per week, some complete with the patches necessary to implement the feature. These feature requests produced a list of 320 requests before 8.10 development even began.

When deciding which features to implement and how they should be implemented, we try to balance backwards compatibility with change. By introducing radical changes gradually, we give *sendmail* sites a chance to prepare for the changes. A combination of a huge user population and 20 years of *sendmail* availability prevents us from doing radical changes without advanced warning. For example, the 8.9 documentation included a notice warning users that configuration file names would be changing in 8.10. Also in 8.10, the LDAP map class will be changed from `ldapx` to `ldap`, thereby dropping the class name’s connotation as an experimental map. The old name will continue to work (and print a warning) in 8.10, but will be removed in a subsequent release.

Some of the other open source mail transfer agents, such as *Postfix* and *qmail*, are not yet so constrained.

### 3.3. Alternatives to Sendmail

At the same time, these other open source MTAs also drive *sendmail* development. The open source alternatives, such as *Postfix* and *Exim*, give *sendmail* a (for the most part) friendly form of competition. This competition promotes both innovation and sharing for all of the MTAs. For example, Wietse Venema, author of *Postfix*, not only asked about *sendmail* behavior during his development of *Postfix* (to maintain compatibility), but also made contributions to *sendmail*.

With this mostly friendly competition and cooperation among open source authors, everyone wins. Without multiple open source implementations, there would be no choice for the user, nor much pressure to move the existing implementation forward or adhere to standards issues. With multiple implementations, users are free to choose the open source MTA with which they are most comfortable. Since the MTAs are all based on open standards instead of commercial, proprietary standards, they are able to interoperate and prevent the Internet from becoming proprietary and vendor specific.

## 4. Enter Sendmail, Inc.

As one might imagine, the creation of Sendmail, Inc. represents a major change in the development of the open source version of *sendmail*. Now, there is a commercial entity behind the development—a company that is completely committed to the open source. The development of *sendmail* would have continued without the creation of Sendmail, Inc., but at a slower pace and with fewer resources. Sendmail, Inc. was able to release its first commercial product, Sendmail Pro™ quickly and successfully thanks to the already available and proven open source *sendmail*. In return, Sendmail, Inc. can contribute both financial and human resources to open source development. These contributions can be found in the many places within the company.

### 4.1. Engineering

As detailed in the history section, up until the formation of Sendmail, Inc., all *sendmail* development and support was done by volunteers in their spare time. This development model limited the total energy that could be exerted. Sendmail, Inc. has been able to

create a complete engineering team to work on *sendmail*, including software engineers, quality assurance (QA) engineers, and technical writers.

There are currently two full-time engineers working on the next version of the open source, Gregory Shapiro and Claus Assmann, with more to be added as they can be hired. These new engineers will help deal with the growing complexity of new standards and respond to new user requests as they arrive. Additionally, other engineers are working on commercial products, and selected features from those products are being included in the open source distribution.

The presence of a QA department has an additional impact. Previously, formal testing was minimal; in particular, formal testing tools, such as code coverage tools, were not applied by *sendmail* core developers. The Sendmail, Inc. QA department now provides the first line of formal testing before release to outside testers.

The technical writers provide professional writing and editing resources to improve and expand the available documentation. They will be able to help clean up and augment the available documentation for the open source *sendmail* distribution.

Beyond people, Sendmail, Inc. has made commercial development and software testing tools (such as memory leak detectors and code coverage monitors) available to the engineers. These tools were previously too expensive for the volunteer developers<sup>1</sup>. The company is also able to afford a variety of hardware platforms giving development and QA engineers the chance to test portability in-house before releasing a distribution for testing.

This “hybrid” approach to *sendmail* development introduced some new concepts for the developers, for example, project schedules and structured code reviews. Previously, *sendmail* development was not done according to a schedule and there were no hard deadlines<sup>2</sup>. Releases were made when they were ready instead of on a predetermined date. The addition of commercial influences does not mean that releases will be made before they are ready. Instead, given a future release date, the number of features that can be

---

<sup>1</sup>It would seem a “good thing” if producers of software development tools would consider donating copies of their software to established open source development groups that could not otherwise afford them.

<sup>2</sup>There were targets, for example, *sendmail* releases were often targeted to precede USENIX conferences.

implemented in that time frame are determined. Of course, if there are problems with the release as the release date nears, either those problematic features will be removed before the release or the schedule will slip. No version will be released before it is ready.

As more engineers work on the code, more structured code reviews are planned. Previously code review was done on an ad hoc basis; for example, Allman reviewed most newly contributed code and someone (often Shapiro) reviewed most of Allman's code. Now all code check-ins are mailed to a list of core team members. This mailing has the effect of keeping everyone "in sync" and catching many problems immediately. For wholly new code, more formal code reviews will be instituted.

## 4.2. Support

The support infrastructure of Sendmail, Inc. collects and reports problems, analyzes trends of incoming questions, and provides feedback to the developers. Sendmail, Inc. consultants and engineers now visit customer sites, allowing them to see *sendmail* in use in the field and discuss the customer's needs and expectations. These visits will all lead to improved features and clearer documentation not only for the commercial customers, but for the open source users as well.

Another unexpected benefit has been acceptance by companies that refuse to use "free software" because of fears that they will not be able to get support. Particularly for "mission critical" code such as the mail system, many companies require a commercial support organization that has contractual obligations to answer questions within a certain time frame, and as a last resort, an entity willing and able to stake its reputation on the ability to provide solutions to customers' problems. Sendmail, Inc. can provide guaranteed 24/7 support coverage.

## 4.3. Other Expertise

Sendmail, Inc. also provides specialized resources for handling other tasks unrelated to the software itself, freeing up the developers to do what they do best. Although *sendmail* has always had some form of "marketing" to entice users to upgrade, it has not had a marketing organization to spread the gospel and inform trade press about the features of new versions. That has all changed with the creation of Sendmail, Inc. and a true marketing department. For example, with the formation of the new company and the public beta release of *sendmail* 8.9 in March of 1998, information about the release and its new anti-spam features made the front page of the *New York*

*Times* [Mark98].

With the addition of a business development unit, Sendmail, Inc. is in a better position to partner with other companies to provide enhanced services to the *sendmail* community. For example, third-party commercial virus and spam checkers are planned for availability with *sendmail* 8.10. Since the hooks needed for these third-party plug-ins will be in the open source release as well as the commercial release, open source users will be able to take advantage of these new filtering technologies.

Finally, Sendmail, Inc. provides the legal resources necessary to research and complete the necessary paperwork for the open source distribution, such as licenses and government export approval for features like SMTP authentication and secure SMTP (discussed below).

## 5. The Present: Sendmail 8.10

Even though coding for *sendmail* 8.10 began the first week of February, 1999, plans for the version started even before 8.9 released. The 8.9 release was to be the anti-spam release and there was great demand to get these features out to the users as soon as possible. This time pressure forced us to limit new functionality to spam fighting features and defer others for the next release. We also did not want to obsolete the "Bat Book" [Cost97]; which would be a disservice to our users and the open source customers. From the 320 customer requests, we picked more than 100 features for inclusion in 8.10; another 80 were selected as potential features if time permits in the release cycle.

As with past releases, 8.10 has a "theme". Although many of the other changes are important, we plan to highlight SMTP authentication and a new mail filter API as the premier features for 8.10.

### 5.1. SMTP Authentication

Our hopes are to have SMTP authentication [RFC2554] as part of the 8.10 release. SMTP authentication provides a method for the mail user agent (MUA) to authenticate the user to the mail transfer agent and carry that authentication with the message as it passes between mail servers toward the final destination.

For message submission from an MUA into a site's mail server, SMTP authentication provides a mechanism for recognizing users as trusted for that site.

This feature can be used to allow relaying based on the submitting user instead of the submitting host, a feature especially useful for roaming users submitting mail from untrusted sites.

Although the authentication information is carried in the message envelope until reaching the final delivery host, remote sites should not trust this information as it may have been altered by a “man-in-the-middle” attack. As the RFC notes, SMTP authentication is “generally useful only within a trusted enclave” [RFC2554]; it is not meant as an end-to-end authentication or security mechanism.

Initially, *sendmail* will use the message submission authentication to override the relaying checks. It will also provide the authentication information to user rule sets as macros.

Unfortunately, there is potentially a major road block that would prevent us from including SMTP authentication—the United States government’s cryptography export policy. Although authentication is claimed to be acceptable for exporting, the Bureau of Export Administration may reject our application if the bureau feels the authentication hooks in *sendmail* can be easily converted to provide encryption, even though enabling encryption is not the purpose of the hooks. The definition of “easily converted” is unclear. Surprisingly, the distribution of *sendmail* in source code form hurts our chances of getting approval. Products that do not ship with source code, such as the Netscape<sup>TM</sup> Messaging Server, are able to ship SMTP authentication. In such cases, those products are able to limit the use of the routines to an authentication model that is weak enough to be accepted by the United States government. Also, in binary form, it would be nearly impossible to convert the authentication routines into encryption routines.

Assuming we are able to distribute 8.10 with SMTP authentication, there are still some outstanding issues. In creating the extension, we needed an implementation of the Simple Authentication and Security Layer (SASL) [RFC2222] library to provide the framework for the different authentication methods. The only open source implementation currently available (that we are aware of) is the Cyrus SASL library. Early attempts using this library were not encouraging because of portability and implementation problems. Since we do not currently have the time or resources to create our own SASL implementation, we have decided to use the Cyrus library and contribute all our bug fixes and portability changes back to the Cyrus development group. These changes are being incorporated into the base Cyrus release, making them available to others and providing a stronger

implementation for use with *sendmail* when 8.10 is released. This feedback of changes is another example of Sendmail, Inc.’s commitment to the open source community.

## 5.2. Third Party Mail Filter API

The other major functionality enhancement for 8.10 is a third party mail filter API. This API will allow system administrators and third party companies to provide message filtering using hooks in the *sendmail* code. This new plug-in architecture will allow for better spam and virus monitoring as well as give administrators the ability to accept, reject, discard, modify, or archive messages.

Briefly, *sendmail* will have a compile flag that will implement callouts to user-supplied routines that will be called to process envelope information, headers, and the message body. The filter can request that new headers be added or the entire message body be replaced. In addition, portions of the envelope can be modified—in particular, recipients can be added or deleted. This API provides exceptional flexibility.

## 5.3. Other Noteworthy Changes

Beyond these two new major additions, 8.10 will include many other new features. Although it is impossible to describe them all in this paper, we will mention some of the high points.

**IPv6 support.** In response to network changes, *sendmail 8.10* includes IPv6 support using the interface described in RFC 2553, *Basic Socket Interface Extensions for IPv6* [RFC2553]. This support allows sites that are moving to IPv6 the ability to include *sendmail* in their transition plans and testing.

**New RFC support.** Other RFCs under consideration for possible *sendmail 8.10* inclusion are the message submission protocol [RFC2476], enhanced SMTP status codes to provide more precise error reporting [RFC2034], and anti-spam recommendations [RFC2505].

*Sendmail 8.9* is already compliant with RFC 2505, *Anti-Spam Recommendations for SMTP MTAs*. However, we are investigating including a suggested change so that the MTA can limit the maximum number of messages that can be sent from a particular user in a specified interval. This feature will help to reduce the damage that can be performed by “hit-and-run” spammers.

**Improved virtual hosting capabilities.** The most requested enhancements has been for better virtual hosting support. *Sendmail 8.10* will include better control over the virtual user table, which

provides a domain-specific form of aliasing, allowing multiple virtual domains to be hosted on one machine. A new class is available for triggering virtual user table lookups to match the functionality of the generics table, the feature used to rewrite local addresses into a generic form. “Plus detail” information, the portion of the mail address used to carry additional information about the user address that precedes the plus sign (for example, `user+detail@host`), will also be made available for both generics and virtual user table lookups.

Additionally, 8.10 maintains information regarding the incoming connection in a new macro. For example, hosts having multiple IP addresses on different virtual interfaces always advertise themselves as the primary host name in 8.9. In 8.10, they will be able to identify themselves as the virtual host throughout the transaction. The SMTP greeting and `Received:` headers will use the virtual host name and outgoing IP connections will be bound to the address of the customer instead of the hosting ISP (so the “next hop” SMTP server will log the appropriate host name in its `Received:` lines).

For sites providing queueing services, 8.10 will offer a new mailer flag for queueing mail until delivery is explicitly requested via either a queue run with pattern matching (`-qR`, `-qS`, `-qI`) or via ETRN, the SMTP service extension for remote message queue starting [RFC1985]. This feature provides better support for ISPs that provide queueing for dial-up customers, as queue runs are no longer held up waiting for the dial-up server connection attempt to time out.

**Improved anti-spam features.** To allow users more fine grain control, 8.10 introduces more detailed specification for the access database. Tags on the key of access database entries can limit the lookups to specific anti-spam checks. For example, specifying `To:friend.example.com` instead of `friend.example.com` in the access database, allows relaying *to* `friend.example.com` without permitting mail relaying *from* that site.

A new DNS-based blacklist feature (`dnsbl`) supersedes the Realtime Blackhole List (`rbl`) feature available with 8.9. The new feature takes the name of the blacklist server as well as an optional rejection message. The blacklist server is queried with the IP address of each incoming connection and, if the query is successful and the IP address is blacklisted, the connection is rejected. This new feature can be included multiple times to allow sites to subscribe to multiple servers. `FEATURE(`dnsbl`)` replaces `FEATURE(`rbl`)` to prevent the possible confusion between the Realtime Blackhole List and other DNS based blacklist servers.

Other new anti-spam features include `FEATURE(`require_fqdn`)`, which requires a fully qualified domain name for sender addresses unless the connection comes from a local system, and `FEATURE(`relay_mail_from`)`, which allows relaying if the mail sender is listed as RELAY in the access map.

The ability to delay anti-spam checks until the SMTP `RCPT` command has been added using `FEATURE(`delay_checks`)`. This feature allows sites to permit mail to certain addresses, such as postmaster, regardless of the results of other anti-spam checks.

**New macros, rulesets, and options.** The 8.10 release also introduces new named macros and rulesets for controlling other facets of the daemon. Examples of the new macros include `${rcpt_mailer}`, `${rcpt_host}`, and `${rcpt_addr}`, which represent the resolved triplet that delivers the mail to this recipient. These macros can be used to simplify matching in custom `check_*` rulesets. Three new ruleset calls, `check_etrn`, `check_expn`, and `check_vrfy` have been added to restrict the ETRN, EXPN, and VRFY SMTP services. Instead of globally turning these services off via the `PrivacyOptions` option, administrators can now use the rulesets to allow these commands for certain sites.

New options have been added for general mail server policy and protection. These options include the popular `MaxHeadersLength` and `MaxMimeHeaderLength` options, which protect against “denial-of-service” attacks and buffer overflows in some MUAs.

**Better LDAP integration.** The 8.10 release offers tighter integration with the Lightweight Directory Access Protocol (LDAP), which has proven to be the directory service of choice at many sites. Support for multiple entry/attribute LDAP value searches, LDAP authentication, and LDAP-based alias maps will appear in *sendmail 8.10*. We are also monitoring the IETF LDAP Schema for E-mail Routing “birds-of-a-feather” group for a standard schema for alias specification using LDAP [LASER].

**Improved performance.** In an effort to improve *sendmail*’s performance, 8.10 includes code donated by Exactis.com (formerly InfoBeat™) that provides support for multiple queues in *sendmail*. The new Exactis.com donation also includes code that extends the queue file name, making it unique for a 32-year period. This change reduces the amount of file locking and renaming necessary for instantiating a queued message. In addition, the new queue file naming

system makes it possible to move items between different queues easily and quickly.

Exactis.com also donated the code necessary to implement memory-buffered files on systems that include the Chris Torek `stdio` library, such as the BSD family. If your operating system can take advantage of this new code, *sendmail* will be able to reduce file system overhead by not creating temporary files on disk. In combination with the new queue file naming system and multiple queue support, file system bottlenecks will be greatly reduced.

**Features from Sendmail Pro.** Beyond funding for development from Sendmail, Inc., the open source version also benefited by receiving new MTA features from Sendmail Pro, the commercial product. These changes were released in open source even before Sendmail Pro was released<sup>3</sup>. Two of these changes are new daemon control functionality and trusted user support.

The first, new daemon control functionality, allows an external program to control and query status from the running *sendmail* daemon via a named socket, similar to the `ctlinnd` interface of the INN news server [Salz92]. Although only a few commands (`restart`, `shutdown`, and `status`) are available in this first version, the framework is in place for extending this functionality to control and query different facets of the daemon. Since access to this interface is controlled by the UNIX file permissions on the named socket, the file permissions provide administrators a means of controlling the daemon via external interfaces without requiring root privileges. A Perl program (`contrib/smcontrol.pl`) is provided in the distribution as example code to take advantage of the control socket.

The new `TrustedUser` option allows the administrator to specify a user name that will be considered equivalent to the superuser for permission checks and other operations normally reserved only for root. For example, the `TrustedUser` is allowed to start the daemon as well as own maps, files, and directories without *sendmail* marking them as untrusted. This change is another step in the migration toward a *sendmail* daemon that does not heavily rely on superuser privileges.

**Consistent file names.** As of 8.10, the default location for all *sendmail* configuration files will be `/etc/mail/`. This change avoids sprinkling

---

<sup>3</sup>They were disabled by default as they had not been fully tested at the time of the open source release.

potentially dozens of files in `/etc` with obscure file names, such as `sendmail.cw` (now known as `/etc/mail/local-host-names`), and allows that directory to be owned and managed by the user specified in the `TrustedUser` option. The files affected include maps, aliases, and classes, as well as the error header, help, service switch, and statistics files.

Although the new file names will make configuration and support easier in the future and users were warned of the upcoming change in 8.9, this change will probably be the most traumatic 8.10 change for users upgrading from earlier versions.

**Beyond the MTA.** Outside the *sendmail* MTA itself, the open source distribution includes other utilities, such as *mail.local*, the local delivery agent; *makemap*, the map generation tool; and *praliases*, the tool that converts an alias database back to its textual form. These utilities have also had minor updates to improve ease of use. Although invisible to the end user, code sharing between *sendmail* and these utilities has increased by using portability and utility libraries. By sharing code and breaking the utility routines out of the MTA, 8.10 moves us a step closer to splitting up the monolithic *sendmail* process into multiple programs in future releases.

A new enhanced version of the *vacation* auto-responder is a standard part of the *sendmail* distribution beginning with 8.10. Some of the new features planned for the revised *vacation* include new command line options for specifying alternate databases and alternate messages, as well as a method for getting the sender out of band. The new *vacation* will also support an exclusion list of addresses to which an automatic response should not be generated.

## 5.4. Maybe Next Time

As with any large software project, there are enhancements we had planned on including in 8.10 but were unable to tackle because of resource constraints. At the current time, the two biggest casualties were Windows NT<sup>TM</sup> portability and support for secure SMTP (i.e., encryption) [RFC2487]. While we continue to design with Windows NT portability in mind, the extensive changes required have lead us to postpone this change until a future version.

**Secure SMTP with TLS.** Although secure SMTP is an extremely important feature, arguably just as important as secure web service, the United States government is not expected to allow release of the source code for an encrypting mail server to the world.



It is unfortunate that even though this encryption is widely available in other countries and freely available for download from international servers, the United States still has not recognized that the people being hurt most by these export restrictions on encryption are its own citizens and businesses.

Encryption patches for *sendmail* are available from one of a number of sites outside the United States. As an example, one can look at the *ssmail* patches at <http://www.home.aone.net.au/qualcomm/> [Rose99]. However, this patch does not use the published TLS extension<sup>4</sup>.

We will continue to investigate methods of making secure SMTP with TLS available for *sendmail*. For example, we might produce a “domestic” version of *sendmail* with TLS.

## 6. The Future

There remain several major factors to research and goals to accomplish in future versions. *Sendmail* will need to move toward a threaded model to improve portability for Windows NT. This change will require significant changes to the MTA in both its use of global variables and memory management. Any services, such as DNS and system libraries, that *sendmail* uses will also need to be thread safe. This change may improve or may degrade performance for UNIX systems depending on the thread implementation of the operating system and how it compares to forking, which has become quite efficient on some systems.

A popular trend in newer open source MTA implementations has been to break up the tasks into separate programs. We will be studying the performance trade-offs of making these changes to *sendmail* and breaking tasks off as appropriate. This approach has its benefits as “[i]t has been observed that one of the great successes of UNIX is that each tool does only one job, and therefore can do that job well” [Allm85]. It will also allow us to improve security by securing smaller portions of privileged code.

As we make *sendmail* portable to non-UNIX platforms, we will have to reconsider the I/O subsystem. For example, Windows NT and BeOS<sup>TM</sup> sockets do not have the same semantics as UNIX

---

<sup>4</sup>This limitation may not be a disadvantage; the *ssmail* authors argue that the overhead of TLS is too high for routine use.

sockets. In particular, Windows NT does not have the *fdopen*(3) call, and BeOS sockets are not inherited by forked children. We expect to hide most of the I/O behind another compatibility layer, possibly *sfio* [Korn91].

After 8.10 is released, we expect to do considerable work on performance enhancements and tuning, including memory pools for more efficient memory allocation, support for threaded delivery, and the use of shared memory for saving long-term state.

## 7. Summary

As *sendmail* development continues, it is affected by four driving forces: continually changing network requirements, user requests, available development resources, and competition. None of these factors are particularly new to any popular network server software, but the substance of these factors for *sendmail* are unique. The paper has laid out these factors both historically and in the present, including some of the new features for 8.10 brought about by these four forces.

The most notable event in the evolution of *sendmail* development is undoubtedly the creation of Sendmail, Inc., a “hybrid” business model company producing both the open source and commercial versions of *sendmail*. Sendmail, Inc. helps drive network changes by participating more fully in the IETF. More directly, Sendmail, Inc. provides far more development resources—in the form of funding, people, and tools—to the *sendmail* open source than were previously available. For example, the company has paid for conference calls between members of the Sendmail Consortium and plans to host meetings for the group.

This arrangement benefits both the company and the open source distribution. The open source gains new features and enhancements, while the commercial products reap the benefits of an active open source community contributing both new ideas and testing.

The future promises some exciting times for both the open source distribution of *sendmail* and the commercial products as both grow together.

The latest open source version of *sendmail* is available from <http://www.sendmail.org/>. More information about Sendmail, Inc. can be found at <http://www.sendmail.com/>.

## 8. Acknowledgments

The authors would like to thank David Conrad of the Internet Software Consortium, Sendmail, Inc.'s Mark Delany and Nick Christenson, and Kirk McKusick for their insights and suggestions regarding the paper. We also appreciate the hard work of Claus Assmann, Katie Calvert, and Cathe Ray, for reviewing and improving the paper. Finally, we would like to offer our sincere appreciation to the members of the Sendmail Consortium for all of their efforts and contributions to *sendmail*.

### REFERENCES

- [Allm85] Allman, Eric, and Miriam Amos, *Sendmail Revisited*, Proceedings of the Summer 1985 USENIX Conference, pp. 547-555, 1985.
- [Cost97] Costales, Bryan, and Eric Allman, *sendmail*. O'Reilly & Associates, Inc., Second Edition, 1993.
- [Korn91] Korn, David G., and Kiem-Phong Vo, *Sfio: Safe/Fast String/File IO*, Proceedings of the Summer 1991 USENIX Conference, pp. 235-256, 1991.
- [LASER] Freed, Ned, *LDAP Schema for E-mail Routing (laser) bof*, Meeting Report, Proceedings of the Forty-Third Internet Engineering Task Force, December, 1998.
- [Mark98] Markoff, John, *Battling Junk Email May Become Easier*, New York Times, March 17, 1998.
- [MSGFMT] Resnick, Peter W., *Internet Message Format Standard*, Internet Draft draft-ietf-drums-msg-fmt-07, January, 1999.
- [RFC821] Postel, Jonathan B., *Simple Mail Transfer Protocol*, RFC 821, August, 1982.
- [RFC1985] De Winter, Jack, *SMTP Service Extension for Remote Message Queue Starting*, RFC 1985, August, 1996.
- [RFC2034] Freed, Ned, *SMTP Service Extension for Returning Enhanced Error Codes*, RFC 2034, October, 1996.
- [RFC2222] Myers, John G., *Simple Authentication and Security Layer (SASL)*, RFC 2222, October, 1997.
- [RFC2476] Gellens, Randall, and John C. Klensin, *Message Submission*, RFC 2476, December, 1998.
- [RFC2487] Hoffman, Paul, *SMTP Service Extension for Secure SMTP over TLS*, RFC 2487, January, 1999.
- [RFC2505] Lindberg, Gunnar, *Anti-Spam Recommendations for SMTP MTAs*, RFC 2505, February, 1999.
- [RFC2553] Gilligan, Robert E., Susan Thomson, Jim Bound, and W. Richard Stevens, *Basic Socket Interface Extensions for IPv6*, March, 1999.
- [RFC2554] Myers, John G., *SMTP Service Extension for Authentication*, RFC 2554, March, 1999.
- [Rose99] Bentley, Damien, Greg Rose, and Tara Whalen, *ssmail: Opportunistic Encryption in sendmail*, Draft.
- [Salz92] Salz, Rich, *InterNetNews: Usenet transport for Internet sites*, Proceedings of the Summer 1992 USENIX Conference, pp. 93-98, June 1992.
- [SMTPUPD] Klensin, John C., *Simple Mail Transfer Protocol*, Internet Draft draft-ietf-drums-smtpupd-10, February, 1999.