
序

sendmail 這本書之所以會發行第二版的主要原因是 sendmail 程式發表了 8.8 版。8.8 版的 sendmail 和先前的版本有明顯的差異，可以說是完全改寫。

此外，本書第一版發行之後，來自全球各地讀者的回饋也說明了原書還有許多值得改進的地方。例如，有些讀者覺得原書對於實際的工作助益不大，而且範例組態檔缺少註解；還有一些讀者提供了許多建議，目的是讓原書更清晰易解。有些建議似乎不是很重要，有些則不然；例如，選項的處理部分就需要大量改寫。第二版和原書最明顯的不同，是增加了節次的編號，並加入了運用該編號的交互參照。

序

腓尼基 (Phrygia) 國王哥第斯 (Gordias) 曾打了一個糾纏繁複的結，無人能解。那個哥第斯結 (Gordian Knot) 一直維持原有的模樣，就像故事所說的，直到亞歷山大大帝的到來，才以一種史無前例的方式將它解開。如果 sendmail 就像那個結，能夠以全新觀點的快速一擊，迅速破解，該是多麼的美好！可惜呀，沒辦法，我們必須採用比較平凡的途徑。所以本書將用辛苦的方法來解開這個結。

但你可能會問：「何必要如此辛苦？sendmail 不是在電腦時代的初期就誕生了？難道現在不能用一種更新、更好、更酷的東西來取代它嗎？」不行。時間使得 sendmail 更成熟也更穩定。sendmail 歷經了長時間的考驗，它不僅僅是一個程式，還是一種哲學：它是一種具有高度彈性與可調整性的通用型網路郵件繞送工具，不論網站規模的大小，也不論郵件功能是複雜或簡單，sendmail 對於你的需求可說是一應俱全。

sendmail 優越之處，卻也是其弱點之所在，它的組態設定極為複雜，sendmail 程式非常難設定，甚至難以理解。舉例來說，它的組態檔實在是嚇人。但你也不必沮喪，只要手上有這本書，你應該可以依需要來設定 sendmail 的組態，所謂的 sendmail 大師，也將隨之走入歷史。

歷史

sendmail 最初是由 Eric Allman 所設計出來的，當時他還在加州大學柏克萊分校唸書以及任職。該校有一部電腦（Ingres）連上了 ARPANet，它是 Eric 參與 INGRES 專案時所採用的主機；另一台電腦（Ernie CoVax）供 Berkeley UNIX 專案使用，正要進行 UUCP 的實驗。這些電腦（和校園其它的電腦一樣）乃是透過 Eric Schmidt 所架設的一種低成本網路互相連接，稱為 BerkNet。無論是在 ARPANet、UUCP 或是 BerkNet，都有軟體可以在個別的網路之間傳送郵件，卻沒有軟體可以在這三者之間傳送郵件。

由於各種的通訊協定紛紛出籠，以及預期網路數量的暴增，促使 Eric 寫出了 delivermail，它就是 sendmail 的前身。在 1979 年，4.0 與 4.1 版的 BSD UNIX 納入了 delivermail，很不幸的，delivermail 缺乏足夠的彈性來處理複雜的郵件繞送需求，而這些情況在實際的運用中層出不窮。事實上，它最大的弱點在於將組態設定與程式編譯在一起。

1980 年，ARPANet 放棄了 NCP（Network Control Protocol，網路控制協定），而改用 TCP（Transmission Control Protocol，傳輸控制協定）。這項劃時代的改變，使得主機的數量可以從 256 台增加到十億台以上。另一項改變，是將主機名稱的命名方式由扁平式（flat，如 MIT-XX）改為階層式（hierarchical，如 XX.MIT.EDU）。在做這些改變之前，郵件是透過 FTP（File Transfer Protocol，檔案傳輸協定）傳

送的，之後則是透過一種新的協定傳送，稱為 SMTP (Simple Mail Transfer Protocol, 簡易郵件傳輸協定)。這些發展和改變都不是一夕可成的，而是漸進式的。多年之後，當大部分的網路都已改用 TCP 協定時，仍有一些網路還在執行老舊的 NCP 協定。SMTP 協定本身也經過多次的修改，最後才能有今日的模樣。

因應這種種的改變，Eric 將 delivermail 發展成為 sendmail。為了確保郵件在網路間傳送時都能遵循各網路所要求的規則，Eric 採取了自由開放的方式來修改地址資訊，使其能滿足各種嚴苛的要求，以免郵件被莫名其妙地退回。舉例來說，當時 UUCP 的郵件完全沒有標頭 (header)，所以 sendmail 必須無中生有，替郵件自動產生出標頭來。

第一隻 sendmail 程式出現在 4.1c 版的 BSD (第一個支援 TCP/IP 的 Berkeley UNIX 版本)。從第一版到現在【註】，Eric 持續地改進 sendmail，最初在柏克萊大學，然後在 Britton Lee，後來又回到柏克萊，現今則在 InReference 公司。目前 sendmail 是 8.x 版 (或簡稱 V8)。V8 經過大幅度的改寫，包括了許多錯誤修正與重要的功能補強。

Eric 不是唯一在進行 sendmail 改版的人。1987 年，瑞典 Linköping 大學的 Lennart Lovstrand 為 BSD 第五版的 sendmail 發展了 IDA (Insitutionen for Datavetenskap 的縮寫) 增強版本，不僅加入不少的新功能 (例如對 dbm 檔案的支援以及將標頭與信封分別改寫)，也修正了不少錯誤。到了 90 年代，IDA 出現了兩種衍生版本。

Neil Rickert (北依利諾大學) 與 Paul Pomes (依利諾大學) 接管了 IDA sendmail 的維護工作。藉由來自全球的協助與貢獻，他們的版本 (UIUC IDA) 像是 Lennart Lovstran 初期工作的延續。Neil 將注意力放在錯誤修正並強化組態檔，使其成為目前以 m4 為基礎的形式。Paul 負責維護程式碼，持續加強功能並修正錯誤。一般來說，他們的版本較龐大、野心更大，且具有高度的可攜性。它成功地解決了許多複雜的郵件繞送問題。

註 在 82 到 90 年間有較長的時間斷。

另一個 ID A sendmail 的變體版本是由 Paul Vixie (當時他在 DEC 公司) 所發展, 稱為 KJS (King James sendmail)。它是由 Lenart Lovstrand 最後一版的 IDA 小幅度修改而成。KJS 的焦點是放在程式碼的改進, 而非組態檔的變更。

除了這兩種主流分支以外, 許多廠商也根據本身的特殊需求修改了 sendmail。Sun 對 sendmail 做了許多改進, 包括支援 nis 與 nisplus 的對映表; HP 也做了許多不錯的加強, 包括對 BITMIME 的支援。

sendmail 版本的暴增導致了嚴重的混亂。每個版本對問題的解決方案, 很不幸地, 對其他版本而言都無效; 除此之外, 也造成組態檔不具可攜性, 許多特殊功能無法彼此分享。

1994 年, Eric 開始開發 V8.7 sendmail。V8.7 是第一個突破多年傳統, 具有更明顯改變的版本, 它引進了多字元 (multicharacter) 選項與巨集名稱, 增加了使用 -bt 模式操作的新交談式命令, 並修正了先前版本的許多問題與缺陷; 最重要的, V8.7 正式地將 IDA、KJS、Sun 與 HP 等版本大部份的優點收錄其中, 並符合網際網路工程任務小組 (Internet Engineering Task Force, IETF) 的最新標準。Eric 於 1996 年發展 V8.8 sendmail。這一版持續著 V8.7 的發展方向, 增加了許多使用者渴求的特殊功能與選項, 並加強了安全性的功能。因為 V8.8 是目前 sendmail 的正式版本, 所以本書只針對它進行說明。

Eric Allman 的話

我必須承認, 我對 sendmail 的成功非常驚訝。它的成功並不是因為有龐大的行銷組織, 也沒有巨額的廣告和研發預算。我認為是下面三個原因。

第一, sendmail 面對即使是非常糟糕的郵件, 選擇了一種試著接受並將其整理、遞送的方式, 而不會因為這些郵件不符合某些協定便拒絕接受。我覺得這點非常重要, 因為我那時正嘗試建立 UUCP 到 ARPAnet 的開道。當時 ARPAnet 的規模小的可憐, UUCP 則是一團混亂 (有人認為現在還是差不多), 而 UNIX 的 mail 程式通常對標頭並不瞭解。這項工作極為困難, 但畢竟其目的是為了通訊, 而非賣弄學問。

第二，我約束自己只處理繞送的功能，而不撰寫使用者代理程式（user agent）或遞送的後端程式（back-ends）。這違背了當時的主流想法。當時的主流派一致認為，繞送邏輯、本地端遞送和一般的網路程式碼，都應該直接放入使用者代理程式中。但我仍然採用自己的作法，讓人們能很快地與新型網路相結合。

第三，sendmail 的組態檔具有足夠的彈性，可適應快速變遷的世界：80 年代已可看到新協定、網路與使用者代理程式的增長。

當然了，它是免費的，可適時地使用並完成任務，也都是成功的因素。

這是個複雜的世界，所以 sendmail 的設定也很複雜；這是個動態的世界，所以 sendmail 的設定也是動態的。或許有一天，sendmail 和 X11 都會銷聲匿跡，但我不會因此而停下腳步。在此之際，希望這本書對你有所助益。

當我開始校閱 Bryan 的草稿時，我曾經避免對 sendmail 做出任何重大的修改。但後來我開始讀到關於一些各式各樣的小錯誤與討厭的內容時，修改起來似乎也不難，所以我開始做些小修正，然後是大修正；然後我研究了 RFC1123，並將 sendmail 的規格全面更新，除了處理一堆八位元的問題之外，也增加了 ESMTP 的功能。如果說，本書和 V8 sendmail 具有相輔相成的關係，似乎是挺公允的說法。

未來的演變

隨著網際網路的快速成長，電子郵件已成為人們普遍接受的一種通訊形式，這確定了未來 sendmail 仍將持續發展。就在 V8.8 發表之前，Eric 已經在思考有關 V8.9 的事情。【編註】想一瞥未來的演變，請參考原始程式碼中的 KNOWNBUGS 檔案。請注意，其中有些問題是相當棘手的。

此外，不斷從 Eric 腦後所滲出的，是他所謂「Sendmail 的孩子」（Son of Sendmail, sos），目標是要為長遠的未來打算，將 sendmail 徹底改寫。密切關心原始程式發展的讀者可能已經注意到，許多函式不斷重新設計，以適應未來龐大的改寫工程，也就是更為模組化。

本書的架構

本書分為四大部份，每一部份都採取一種特別的角度，企圖呈現出 sendmail 的整體面貌。

第一部份 基礎篇

這部份是個自我學習的單元，可做為初學者上手及按部就班的指引，也可提供高階使用者作為複習。第一章到第四章，對電子郵件作了一番概括性的介紹，並討論 sendmail 的角色功能、運作方式與各個組成部分。第五章到第十五章詳細地檢視組態檔。其中我們發展了一個迷你的組態檔，可適用於一些用戶端的工作站。第十六章與第十七章總結此基礎篇，將一些零碎的枝節整理好，並轉換至較複雜的組態檔。【編註】

第二部分 安裝篇

這個部分涵蓋了 sendmail 的編譯與安裝。第十八章說明如何從原始程式碼編譯及安裝 sendmail。第十九章解說如何利用 m4 (1) 組態技術產生組態檔。第二十章則藉由 checkcompat() 常式的實例說明作一總結。

第三部分 管理篇

這部分針對較具經驗的使用者，說明 sendmail 的一般管理事項。第二十一章討論 DNS，並特別解說 MX 的記錄項。第二十二章說明如何保護你的機器避免遭受入侵。第二十三章至第二十六章則完成對佇列、別名、郵寄名單、記錄與統計的細節說明。

編註 第一到第十七章收錄在《sendmail 基礎篇》一書，而本書則涵蓋其餘篇幅（第十八到第二十八章）。

第四部分 參考篇

第四部分是全書的核心，列出了重要的技術參考文件，提供 sendmail 的完整細節，甚至可能超過你實際所需。每一章都著重在 sendmail 或其組態檔的某一特別層面。例如，三十一章詳細地定義巨集，並包含以字母順序排列的參考資料。第三十七章則說明全部的除錯選項。

本書的對象

這本書主要是給負責管理電子郵件的系統管理員閱讀的。但並非所有的 UNIX 系統都是由管理員所管理，許多反而是由程式設計師、網路工程師，或甚是經驗不足的使用者所管理。我們希望這本書能滿足所有的人，無論其經驗與背景。

初學者應該從第一部份開始看起，真正有必要時，才跳到後面的章節。

剛開始管理系統的人，可能必須從第一部份的基礎篇開始，然後閱讀第三部分，以便瞭解如何管理 sendmail。注意，第二與第四部分會對許多繁雜的問題提供解決方案，這些問題可能在其它地方無法獲得解決。

有經驗的系統管理員若想安裝及管理 V8 sendmail，應該先研究第二與第三部分，獲取了必備的背景知識之後，再閱讀第四部分。

精通 UNIX 與 sendmail 的專家會發現，第四部分是非常實用的（即使是 Eric 本人，都在桌上放了一份）。這個部分將 sendmail 每一個不可思議的細節依照字母順序列出。例如，有一章專門研究選項，列出了所有的選項並加以詳細解說。

無論你的專業技術背景屬於何種程度，依據本書設定的篇幅和份量，使得我們必須假設，你對 UNIX 系統的例行工作已經相當熟悉；如果不是，你還得回頭去研究 UNIX。

UNIX 與 sendmail 的版本

本書大部分是根據 BSD UNIX 或其變體版本（例如 SunOS 4.x）的系統環境來解說 sendmail。若在 AT&T System V (SysV)，例如 Sun 的 Solaris 2.x，環境中的情況有所不同時，我會說明其中的差異。【編註】

本書的焦點集中在 8.8 版的 sendmail。為了顧及完整性與實用性，我們也會討論 V8.7 或更早期的版本（例如 BSD 上的 V5【註】、IDA 版本，以及早期的 Sun、Ulrix 與 NeXT 版本）。

印刷體裁

本書所使用的印刷體裁如下所示：

斜體字 (*Italic*)

用於名稱，包括路徑名稱、檔名、程式與命令名稱、使用者名稱、主機名稱、機器名稱、郵寄名單的名稱，以及郵件地址。另外也在介紹新名詞與概念時做為強調之用。

固定字寬 (Constant Width)

用於顯示檔案內容的範例與命令的輸出結果。包括了組態檔、郵件訊息檔、shell 命令檔或 C 語言程式碼的範例。只有在顯示內含的空格時，固定字寬的文字才會用引號夾註，例如五個字元的「From」標頭。

單一字元、符號表示式 (symbolic expressions) 與命令列開關都是以固定字寬顯示。例如 `o` 選項表示一個單一字元，規則 `$-` 是一個符號表示式，而 `-d` 則為一命令列的開關。

編註 由於 Linux 兼容了 BSD 與 SysV 的優點，因此也適用本書。

註 之所以會從 V5 跳到 V8，是因為發行 BSD 4.4 UNIX 的管理員希望所有的軟體都是以第八版發行。原本新的 BSD *sendmail* 應該是 V6，在跳到 V8 之前，V6 只發行了 alpha 與 beta 的版本。

固定字寬粗體字 (Constant Bold)

用於命令或其他應該由使用者鍵入之文字的範例。例如 `cat/etc/sendmail.pid` 的字句意指使用者應該完全依照內文或範例所示鍵入 "cat/etc/sendmail.pid"。

固定字寬斜體字 (Constant Italic)

表示可能或將會以特定文字取代的變數。例如在字串 `Snum` 中，`num` 將會是一個由使用者指定數值的整數。在 `error: num` 的輸出結果中，`num` 將會是一個由程式（通常是 `sendmail`）印出的變數值。

粗體字 (Bold)

在內文中，第一次出現的中文專有名稱將以粗體字印刷。

%

表示使用者的 shell。

#

表示 root 所用的 shell。

其他的資訊來源

sendmail 原始程式碼套件中附了兩篇由 Eric Allman 所寫的必讀文件。Sendmail An nternetwork Mail Router 綜覽 sendmail 與其理論基礎。Sendmail Installation and Operations Guide 則提供了安裝的指引與組態檔的詳細說明。許多廠商也提供了線上說明文件，其中可能透露了廠商特別修改的內容，這些將不在本書中述及。此外，如果你有原始程式碼，也請研究一下 `RELEASE_NOTES`、`src/README` 與 `cf/README` 三個檔案。

RFCs

如果沒有 Request for Comments (RFC) 的輔助，你將不可能對 sendmail 有完整的瞭解。RFC 是由網路資訊中心 (Network Information Center, NIC) 的網際網路工程小組 (Internet engineering Task Force, IETF) 所發佈的。這些經過編號的文件，主要是定義 Internet 的協定與運作需求。

RFC821

當 sendmail 在 TCP/IP 網路上將郵件從一台機器傳送到另一台機器時，它使用的是 SMTP (Simple Mail Transfer Protocol)。RFC821 便是定義 SMTP 的文件。

RFC822

將郵件訊息分隔成標頭與本體，以及標頭行的語法及順序，都在標題為「Standard for the Format of ARPA Internet Text Messages」的 RFC822 中有所定義。其中也說明了地址的語法。

RFC819

RFC819 的標題為「Domain Naming Convention for Internet User Applications」，說明目前用來命名主機名稱的階層形式。其中定義了主機名稱必須使用的格式。

RFC976

RFC976 的標題為「UUCP Mail interchange Format Standard」，其中說明了在以 UUCP 相連的機器間傳送郵件的格式。

RFC1123

RFC1123 是 RFC821 與 RFC822 的擴充。它對原有的文件作了若干修正，並刪除一些意義含糊的資訊。

RFC1521 與 RFC1522

RFC1521 介紹與說明「Mutipurpose Internet Mail Extensions，簡稱 MIME」的標準。MIME 提供了一種將非文字資料（譬如影像、聲音與影片）放入普通電子郵件訊息的方法。

RFC1651、RFC1652 與 RFC1653

RFC 1651 說明了 SMTP 的通用擴充機制，稱為 ESMTP。RFC1652 說明了傳送八位元資料的一個擴充集，稱為 8BITMIME 擴充集（參閱 § 34.8.22）。RFC 1653 說明了訊息長度宣告的擴充集。

RFC1891、RFC1892、RFC1893 與 RFC1894

RFC 1891「SMTP Service Extension for Delivery Status Notifications」說明 ESMTP 之 RCPT 命令的 NOTIFY、RET 與 ORCPT 擴充集，以及 ESMTP 之 MAIL 命令的 ENVID 擴充集。RFC1892「The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages」說明了 DSN 所需的 Content-Type: 標頭，以及在郵件狀態報告的郵件訊息中，多重成分 / 報告 (multipart/report) 的 MIME 成分。RFC1893「Enhanced Mail System Status Codes」說明了 DSN 狀態編號欄位的意義。RFC1894「An Extensible Message Format for Delivery Status Notifications」針對 RFC1892 所描述之 DSN 回傳訊息的機器可讀取成份，說明其標頭樣式的關鍵字。

其他的參考書籍與問題

DNS 與 TCP/IP 網路通訊兩個主題，在本書中僅略為述及。在一個典型的網站裡，有許多郵件被退回的問題並不是由 sendmail 造成的，而是由這類問題所造成的。網域名稱系統在 Paul Albitz 與 Cricket Liu 所著之第三版的《DNS and BIND》中有詳細的說明。

關於 TCP/IP 的問題，請參考 Craig Hunt 所著之第二版的《TCP/IP 網路管理》。

想知道如何做個稱職的 UNIX 系統管理員 (root)，請參考 Æleen Frisch 所著的《Essential System Administration》。

最後，許多郵件的問題只能由系統管理員自行解決。sendmail 程式是由 root 執行，而且只能由 root 安裝與管理。

致謝

我首先最要感謝的人是 George Jansen，他費時數月，將我原本恐怖的意識以及流於形式的散文，在不損及文意的情況下，轉換為目前適合出版的風貌。他真是一位成就卓越而容易被忽略的英雄，也是一位優秀的編輯。

Jon Forrest 與 Evi Nemeth 對基礎篇章節的寶貴意見，協助我將那幾章大動手術，將焦點集中於更實用的形式。千萬別以為人多壞事，對本書來說，他們的鼎力相助，使得本書更趨於完美。

Cricket Liu 負責審視 DNS 這一章，他發現了一些其他人都沒發現的錯誤。Bruce Mah 與 Sean Brennan 分別是第一版與第二版的實驗品，他們依據初期的草稿安裝並執行 sendmail，找出了一些需要修正的疏失。Gavin Cameron 勇敢地將 checkcompat() 的範例應用在真實世界的環境裡，替我進行除錯。John Funk 好心地讓 alpha 測試版的 sendmail 在 Mercury Mail 正式運作。

不用說，如果不是 Eric Allman 先寫了 sendmail，就根本不可能有這本書。他親自校對每一份初稿，確保技術上的正確、提供極具價值的觀點，並為 sendmail 的問題提出了有趣的解答。

Neil Rickert 也看過第一版所有的草稿。但很可惜，非他本人所能掌控的環境迫使他無法參與第二版的工作。

許多的感謝與讚頌必須歸功於 Tim O'Reilly，因為他同意優先出版本書。他的豐富經驗使得本書形成了今日的面貌。他完全能夠掌握整體的巨觀圖像，並充分了解讀者的感覺。如果沒有他的意見，這本複雜而龐大的書是出不來的。

另外還要感謝 Lenny Muellner，他調校了 troff 的巨集，以滿足這有點獨特之原稿的需求。感謝 Edie Freedman，她寬宏大量地接受我對許多封面設計所表現的情緒。

O'Reilly 的製作人員提供了各種方面的專業協助，而完成了這本傑出的書。為此要特別感謝 John Files 綜理這項計畫，Barbara Willette 對原稿的編輯，Nancy Kotary 對最後成品的幫忙，Kismet McDonough-Chan 在每一個產品階段的協助，Chris Reilley 的製圖，Mary Anne Weeks Mayo 對品質控制的協助，Curt Degenhart、Madeleine Newell 與 Ellie Fountain Maden 的編輯，Seth Maiselin 的索引製作，以及 Danny Marcus 的校對。

批評和建議

我們永遠樂意聽到讀者對出版品的意見，包括如何讓本書可以更好的建議、指正本書的錯誤、或是讀者建議本書往後改版時，應該再加進來的其它主題。以下是本公司的聯絡資料：

美商歐萊禮股份有限公司台灣分公司

電話：(02) 2709-9669 傳真：(02) 2703-8802

Web：<http://www.oreilly.com.tw>

電子郵件：

sales@oreilly.com.tw (業務部)

edit@oreilly.com.tw (編輯部)

bookquestion@oreilly.com.tw (書籍內容的問題)

請以電子郵件的方式與我們聯絡，這會比電話和傳統郵件方便。有興趣為本公司翻譯書籍的眾家高手，可與編輯部聯絡；如果您買到的書有印刷品質上的問題，可以寫信到業務部；若您對書籍內容有疑義，或是發現錯字，請寫信到 bookquestion@oreilly.com.tw 或直接與我們聯絡，謝謝您！

