
序

你可能對網域名稱系統（Domain Name System，DNS）的瞭解還不是很深，但是，只要你曾用過 Internet，就必定用過 DNS。你能收發電子郵件、瀏覽網站，全要靠 DNS 幫忙。

身為人類，我們寧願記憶電腦的名字，而不是它們的位址。Internet 上的每部電腦都有一個 32 位元的位址 - 一個介於 0 與四兆之間的數字【註】 - 要記憶這種數字，對電腦而言是小事一樁，但是人腦可就吃力了。不信邪？找一本電話簿，隨便挑十組號碼出來背背看，不容易吧！而記憶任意十組 internet 位址的難度，差不多就是這樣！

註 未來可能會成為標準的 IPv6，其位址有 128 個位元，換句話說，你的電腦位址可能是個 39 位數的十進位數字。

這只是我們需要 DNS 的部份原因而已。DNS 負責處理“主機名稱”與“internet 位址”之間的對應關係，前者是為了方便人類記憶，後者是電腦實際運作時所用的。事實上，DNS 是在 Internet 傳佈和存取任何與主機相關資訊的標準機制，而並非只有位址而已。幾乎所有與網路相關的軟體，都會直接或間接地用到 DNS，包括電子郵件、遠方終端機模擬（例如 telnet）、檔案傳輸（ftp）、瀏覽網頁（Netscape Navigator 與 Microsoft Internet Explorer）... 等等。

DNS 的另一個重要特性，是它使得主機資訊能散佈到 Internet 上的各個角落。將主機資訊放在某部電腦，只有使用該電腦的使用者能受益，但是 DNS 卻能讓網路上任何角落的使用者都可以從遠端取得資訊。

不僅如此，DNS 還能讓你把主機資訊的管理工作分散到許多站台或組織。你無須把你的資料都集中到某中央主機，或定期擷取“主要”資料庫的副本。你只要確認自己控管的區域——我們稱為轄區（zone）——其名稱伺服器上的資訊符合實際狀況即可。你的名稱伺服器能將你的轄區資料提供給網路上所有其它名稱伺服器。

由於資料庫是分散的，系統需要具備能力為你找到想要的資料，也就是說，系統必須有能力去搜索一些可能會有資料的地點。而 DNS 能導引名稱伺服器在整個分散的資料庫中，找出任何一轄區的資料。

當然，DNS 並非完美無缺，它也有些問題。例如，為了互相備援，系統允許同一個轄區內可以有超過一部以上的名稱伺服器擁有該轄區的資料，在這種情況下，伺服器之間互相複製轄區資料時，就有可能會突然發生資料不一致的狀況。

但是，DNS 最“糟糕”的問題，是儘管它在 Internet 上被如此廣泛地使用，但是提及管理與維護 DNS 系統的文件真的非常少。Internet 上絕大多數的管理者，都必須與廠商打交道，看看他們能提供哪些有用的資訊，或利用郵寄論壇（mailing list）、或 Usenet 新聞群組互相討論交換心得。

缺乏文件背後的意義，是 Internet 上最龐大、最重要、最關鍵服務的知識，竟然像是祖傳秘笈一樣，僅有少數幾位管理者擁有，而獨立的工程師與程式設計師，則必須不斷地去嘗試別人已經犯過的錯、學習可能錯誤的觀念。

這本書正是為了避免這種狀況而寫的。然而，我們瞭解並非所有讀者都有成為 DNS 專家的意願（或時間）。會購買本書的讀者，不外乎是系統管理者、網路工程師、或軟體開發者，不管你是哪一種身份，除了管理網域或名稱伺服器以外，想必你還有許多其它事要做，因此，我們不會建議你把這本書重頭讀到尾，然後成為 DNS 專家。然而，我們將嘗試給你足夠的資訊，讓你能完成你需要做的事；不管你控管的是個小網域、或是個大型的跨國怪物，不管你只負責維護一部名稱伺服器，或同時要駕馭幾百部伺服器，在本書你都可以讀到你需要的知識。

DNS 是個大議題 - 大到需要兩位作者 - 無論如何，我們會盡可能讓本書內容容易讀易懂，並同時兼顧實際與理論。在前兩章，我們將說明 DNS 的理論基礎，並提供足夠的實際資訊。隨後的章節將逐一填補各項議題的細節。稍後看到本書的閱讀指南時，你就會知道，要完成你想做的工作，應該要看哪些章節。

當我們談論到實際的 DNS 軟體時，幾乎都是指 BIND (Berkeley Internet Name Domain) 這套最通行（就我們所知，它同時也是最好的）而且符合 DNS 規格的實作成品。我們嘗試彙整自己用 BIND 管理與維護網域的經驗，去蕪存菁後寫入本書。順便一提，我們所管理的網域，是 Internet 上最大的網域之一（這不是為了自誇，而是希望讀者能信任我們所提供的資訊），在必要時，也會提及我們實際用於管理工作的工具軟體，為了速度與效率，這些程式多半都用 Perl 重寫過。

如果你才剛起步，我們希望這本書能幫你征服 DNS 與 BIND；如果你已經熟悉它們，我們希望這本書能讓你溫故知新。

軟體版本

本書討論新版的 BIND 8.1.2 與較舊的 4.9 版。在寫作本書時，BIND 8.1.2 是最新的版本，不過，被非所有 UNIX 廠商都將此版本移植到他們的作業系統。此外，我們偶而會提及 BIND 的其它版本，尤其是 4.8.3，因為許多 UNIX 廠商仍以此版的移植碼為他們產品的一部份。當我們提到 4.8.3、4.9 或 8.1.2 版特有的功能時，或者某版本比較特立獨行時，我們都會嘗試解釋其中的差異。【譯註】

譯註 其實 BIND 8.1.2 已被發現存在有 Denial of service (癱瘓服務) 的安全漏洞，期間歷經 8.2、8.2-P1、8.2.1、8.2.2、8.2.2-P1 一直到目前的 8.2.2-P7 版本。

本書架構

本書章節的編排方式是循序漸進的。第一和二章，討論網域名稱系統的理論。第三到六章，協助你判斷是否應該設定你自己的網域、從何處著手、應該如何選擇。第七、八、九和十等中間的章節，討論如何維護你的網域、如何設定主機以便使用你的名稱伺服器、如何作好擴充網路的規劃以及如何建構子網域。第十一到十五等後面的章節，則是關於除錯工具和問題的探討以及使用 resolver 函式庫常式來撰寫程式時一些不為人知的技巧。

本書的章節編排如下：

- 第一章，背景介紹

提供簡短的歷史以便透視全貌以及討論 DNS 發展的緣由，然後概略說明 DNS 的工作原理。

- 第二章，DNS 的運作細節

更詳細的討論 DNS 的工作原理，包括：DNS 命名空間的結構、網域以及名稱伺服器。我們同時也會介紹一些重要的概念，譬如：名稱解析以及快取。

- 第三章，從何處著手？

說明如何取得 BIND 軟體（如果你尚未擁有的話）以及一旦取得之後該怎麼辦：如何選用你的網域名稱以及如何跟可以委任網域給你的主管單位聯繫。

- 第四章，設置 BIND

詳細的說明如何設置你的頭兩部 BIND 名稱伺服器，包括：建立你自己的名稱伺服器資料庫、啟動你的名稱伺服器以及檢視它們的運作是否正常。

- 第五章，DNS 與電子郵件

討論與 DNS 之 MX 記錄有關的議題，管理者可以透過 MX 記錄指定代理主機，以便處理特定目的地的信件。這章將會說明廣泛應用在網路以及主機上郵件選徑（mail routing）策略，包括使用防火牆的網路以及無法直接連通 Internet 的主機。

- 第六章，設定主機的組態

解釋如何設定 BIND resolver 的組態。我們還說明了多個主要 UNIX 廠商所提供之 resolver 版本的功能特性，最後當然少不了 Windows 95 和 NT 的 resolver。

- 第七章，BIND 的維護與管理

定期的維護是網域名稱系統能夠正常運作的保證，譬如檢查名稱伺服器的健康狀況（日誌系統所登錄的訊息）以及權威資料檔。

- 第八章，當網域成長時

說明擴展網域的因應計畫，包括：如何擴大網域的容量以及主機遷移、停電和斷線的因應計畫。

- 第九章，子網域的分割與管理

探索成為父網域的樂趣。我們會解釋何時應該變為父網域（產生子網域）、如何為子網域命名、如何建立子網域以及如何看管它們。

- 第十章，進階的功能和安全的管理

介紹一些較不常用的名稱伺服器組態設定項，透過這些設定項的協助，你可以調校名稱伺服器的運作狀況、保護你的名稱伺服器以及簡化管理的業務。

- 第十一章，操作 nslookup 程式

展示 nslookup 工具程式實施除錯作業時的輸入與輸出，其中還包括如何解讀遠端名稱伺服器所提供之不明確資訊的技巧。

- 第十二章，解讀 BIND 的除錯訊息

這章是 BIND 除錯資訊的 Rosetta Stone（羅塞達石）【譯註】。本章應該可以協助你理解 BIND 所送出的除錯資訊，進而讓你對名稱伺服器有更深入的了解。

譯註 Rosetta Stone（羅塞達石）是解釋古埃及象形文字的線索。

- 第十三章，排除 DNS 與 BIND 的問題

先介紹 DNS 與 BIND 許多常見的問題以及解決方案，然後說明數種不常見的、難以診斷的狀況。

- 第十四章，DNS 程式設計

示範如何展寫 C 程式透過 BIND 的 resolver 常式詢問名稱伺服器以及擷取資料；同時也會介紹 shell 以及 Perl 等命令稿的程式設計方式。我們還提供一隻有用的程式（希望你也這麼覺得！）你可以用來檢查自己名稱伺服器的健康狀況以及權威資料的設定是否正確。

- 第十五章，補遺

將一些尚未探究的議題納入本章討論。我們會介紹 DNS 的萬用名稱、透過防火牆連通 Internet 的特殊組態設定、透過撥接網路連通 Internet 的主機與網路、網路名稱的編碼以及新的、實驗性的記錄型態。

- 附錄 A，DNS 訊息的格式以及資源記錄

解析 DNS 之詢問與回應的訊息格式以及廣泛地表列目前所定義的資源記錄型態。

- 附錄 B，Sun 主機如何編譯以及安裝 BIND

介紹在 Solaris 2.X 作業系統上編譯最新版 BIND 的程序。

- 附錄 C，Top-Level 網域

表列 Internet 命名空間目前的頂級（或頂層）網域。

- 附錄 D，gTLDs（.com、.net 和.org）的註冊公司

表列目前提供 gTLDs（.com、.net 和.org）註冊服務的公司。

- 附錄 E，in-addr.arpa 網域的註冊表

位於西半球的人可以利用此表向主管單位 ARIN（American Registry of Internet Numbers）註冊 in-addr.arpa 網域的子網域。

- 附錄F，BIND 名稱伺服器以及 Resolver 的設定項
摘要說明名稱伺服器以及 Resolver 之組態設定項的功能、語法、語意以及曾出現過的章節。
- 附錄G，台灣網路資訊中心網域名稱註冊系統說明
針對台灣的讀者特別提供 TWNIC 的網域名稱註冊說明。

閱讀指南

本書主要是為管理網域或名稱伺服器的系統管理者而寫的，但是，它也提供網路工程師與郵件管理者所需的資訊。雖然全書章節的安排是循序漸進的，但並非每種領域的專家都需要逐一閱讀完所有章節。而且我想，你也不會願意重頭看到最後一章才能找到所需的資訊。我們希望這篇指南能協助你規劃如何閱讀本書。

初次設定網域的系統管理者，應該閱讀第 1、2 章的 DNS 理論，參考第 3 章對如何著手開始規劃、挑選網域名稱的建議，看第 4、5 章說明新網域的初次設定工作有哪些，看第 6 章如何設定主機使用新的名稱伺服器；完成這些工作後，你的網域應該可以順利運作了，但是，我建議你還應該看看第 7 章，學習如何架設多部名稱伺服器，以及如何新增網域資訊。然後跳到第 11、12、13 章，學習如何使用排除故障的工具與技術。

有經驗的管理者可以從第 6 章開始，參考如何在不同的主機上設定 DNS resolvers，從第 7 章學到如何維護網域，第 8 章則提供了如何為網域的成長與變遷預作規劃的建議，對於控管大型網域的管理者們，這一章應該特別有價值。第 9 章說明網域的切割與合併，打算進行這種不尋常大動作的管理者，應該看看這一章的建議。第 10 章討論 BIND 8.1.2 新引進的保全措施，對於有經驗的管理者，這些資訊可能頗為有用。第 11 到 13 章所介紹的故障排除技巧與工具，是高級管理者也應該會認為值得一讀的議題。

如果你的網路並非全天候與 Internet 連線，則應該閱讀第 5 章，看看如何在這類網路上規劃郵件服務，以及第 15 章，學習設定獨立的 DNS 基礎建設。

程式設計師應該閱讀第 1、2 章的 DNS 理論，然後參考 14 章對 BIND resolver 函式庫模組之詳細說明。

不直接參與網域管理的網路管理者，仍然應該看看第 1、2 章的 DNS 理論，然後學習第 11 章提供的 nslookup 使用技巧，以及第 13 章對故障排除技巧的忠告。

郵件系統的管理者應該閱讀第 1、2 章的 DNS 理論，然後到第 5 章看看 DNS 與電子郵件如何共存；此外，瞭解第 11 章介紹的 nslookup 工具，應該能幫你從網域名稱空間中挖出郵件選徑資訊。

任何只是純粹對 DNS 有興趣的讀者，可以看看第 1、2 章介紹的 DNS 理論，然後隨興閱讀你認為有用的章節。

請注意，我們假設讀者已經熟悉基本的 UNIX 系統管理、TCP/IP 網路架構，並能使用 Perl 與 shell 命令稿來設計簡單的程式。如果你不具備這方面的能力，建議你先參考相關議題的書籍。當我們介紹新術語或觀念時，我們將盡量給你一個明確而清楚的定義與解釋。如果可能，我們將從 UNIX（以及實際的世界）舉例說明，幫助你瞭解釐清觀念。

取得範例程式

讀者可從下列網址取得本書的範例程式：

```
ftp://ftp.uu.net/published/oreilly/nutshell/dnsbind/dns.tar.Z  
ftp://ftp.ora.com/published/oreilly/nutshell/dnsbind/dns.tar.Z
```

本書的中文網頁也提供下載服務：

```
http://www.oreilly.com.tw/Chinese/network/dns&bind.html
```

不管你用哪一種方式取得，都可以用下列指令解開檔案：

```
% zcat dns.tar.Z | tar xf -
```

若你的系統是 System V，可能必須改用下列指令：

```
% zcat dns.tar.Z | tar xof -
```

如果你的系統沒提供 zcat 指令，就老老實實用 uncompress 與 tar 指令分兩個步驟解開它吧。

印刷體裁

我們採用下列字型與格式印刷 UNIX 指令、工具程式和系統呼叫：

- 組態檔與 script 檔的內容以定寬字型印刷：

```
if test -x /etc/named -a -f /etc/named.conf
then
    /etc/named
fi
```

- 示範人機互動時，一樣以定寬字型印刷，但是會把使用者動手輸入的字樣加粗顯示：

```
% cat /etc/named.pid
78
```

- 對於需要 superuser (root) 權限才能執行的指令，我們以 # 提示符號表示：

```
# /etc/named
```

- 在內文中，第一次出現的專有名詞也會以粗體字印刷。

