DNS HOWTO 中譯版

作者: Nicolai Langfeldt *janl@math.uio.no*
譯者: *Asd L. Chen* & *C.W.Huang*

v1.4.2, 1 January 1998 翻譯日期: 8 January 1998

如何成爲一個稱職的小型 DNS 管理者 ·

目錄

1 前言 **1**
- 1.1 版權宣告 1
- 1.2 感謝與求助 1
- 1.3 題獻辭 1

2 簡介 **1**

3 暫存專用名稱伺服器 **2**
- 3.1 起始 named 5

4 一個簡單的領域 **6**
- 4.1 先來一點純理論 6
- 4.2 我們自己的領域 9
- 4.3 繼續深入 15

5 眞實領域的例子 **15**
- 5.1 /etc/named.boot (或 /var/named/named.boot) 15
- 5.2 /var/named/root.cache 15
- 5.3 /var/named/zone/127.0.0 16
- 5.4 /var/named/zone/land-5.com 17
- 5.5 /var/named/zone/206.6.177 18

6 維護工作 **19**

7 Bind version 8 **20**

8 撥接連線的自動設定 **22**

9 常見問題解答 **24**

10 如何成爲一個大型的 DNS 管理者 **25**

1 前言

關鍵字: DNS, bind, named, dialup, ppp, slip, Internet, domain, name, hosts, resolving

1.1 版權宣告

1.2 感謝與求助

我要向無數次閱讀這份文件的草稿並提供許多有用建議的 Arnt Gulbrandsen 表達感謝之意· 我也要感謝那些將建議寄給我的人·

這份文件將永遠不會有完成的一天，請把你的問題以及成功的經驗寄給我， 這可以使它成爲一份更好的說明文件·所以請把錢，評論以及/或是問題寄給 *janl@ifi.uio.no*· 如果你寄信給我並希望得到回答請表現基本的禮貌，確定回信地址是正確而且可以運作的· 還有，寫信給我前請先閱讀 9 (FAQ) 一節·

如果你想翻譯這篇 HOWTO 請與我聯絡，這樣我才知道我的文章被變成何種語言 :-).

1.3 題獻辭

謹以這份說明文件獻給 Anne Line Norheim Langfeldt·雖然她可能永遠都不會閱讀這份文件，因爲她不是這類型的女孩·

2 簡介

這是什麼以及這不是什麼

首先，DNS 就是領域名稱系統(Domain Name System)· 爲機器命名的規則以及將這些名稱轉換爲 IP 數字的軟體· 這份 HOWTO 文件說明了如何用 Linux 定義這樣的對應· 對應指的是這兩件事的聯繫，機器名稱，像 `ftp.linux.org`， 以及機器的 IP 數字，像是 `199.249.150.4`·

DNS，對新手(你；－)而言，是網路管理中比較晦暗不明的地方之一· 這份說明文件將會試著讓幾件事情能清楚一點·它描述如何設立一個簡單的 DNS 名稱伺服器· 開始先說明暫存專用(caching only)伺服器，然後是爲一領域設立主要(primary)名稱伺服器· 至於更複雜的設定你可以參考這份文件的 9 (FAQ) 一節· 不用說你將得閱讀眞正的文件· 我將會在 10 (最後一章) 回過頭來說明這裡所謂眞正的文件包含有哪些·

在你可以開始進行這項工作之前你應該要先配置好你的機器以便可以從遠端簽入它以及由它對其它機器作遠端簽入， 並且使所有各種對網路的連線都能成功地進行，尤其你應該要能夠對 `127.0.0.1` 進行遠端簽入並且進入你自己的這台機器(現在就試試!)· 同時你也需要一份運作良好的 `/etc/host.conf`, `/etc/resolv.conf` 以及 `/etc/hosts` 檔案作爲開始· 因爲我將不會在這裡解釋它們的功能·如果你還

沒有完成所有這些設定並使其運作那麼 the networking/NET-2 HOWTO 會解釋如何設立它們·閱讀之·

果你使用 SLIP 或者是 PPP 那麼你得要能讓它運作·若是它還不能運作的話閱讀 *PPP HOWTO*·

當我說‘你的機器’時我的意思是你正要嘗試在上面設立 DNS 的那台機器· 不是任何可能在你網路環境裡的其它機器·

我假設你並不是處於任何會阻礙名稱查詢的防火牆後面·如果你是那麼你將會需要特別的設定， 參見 9 (FAQ) 一節·

在 Unix 上的名稱服務是藉由一支稱爲 `named` 的程式來執行的· 這是屬於 bind 套件的一部份，這個套件是由 Paul Vixie 爲網際網路軟體集團(Internet Software Consortium)所協調發展的· 大部分的 Linux 發行套件都包含有 `named`，而且通常安裝於 `/usr/sbin/named`· 如果你的系統有個 `named` 那麼你大概可以直接使用它；如果沒有的話那麼你可以從某個 Linux ftp 站取回一份二進位執行檔，或是從 *ftp.isc.org:/isc/bind/src/cur/* 目錄裡取得最新最棒的原始程式碼· 本文件是說明 bind version 4 的·如果你安裝 version 8 你要自己想辦法·稍後有一節可能會對你有些幫助·

DNS 是個以整個網路爲範圍的(net-wide)資料庫·要小心你放進裡面的資料· 如果你放些廢物進去，你，以及其它人都將會從中取出些廢物· 保持你 DNS 的整齊一致那麼你將能從中取得良好的服務·學習去使用它，管理它，追查它的錯誤那麼你將會是另一個保持網路免於因爲管理不善而效率低落的好管理者·

在這份文件裡我斷然地敘述幾件並不完全是眞實的事情(雖然它們至少有一半是眞的)· 這全都是爲了簡化·如果你相信我所說的那麼這些大概就能運作·

要訣: 如果你已經擁有我要你去改變的檔案，那麼所有這些檔案都要先做備份， 這樣一來如果經歷這些改變之後沒有東西會動的話你可以回復到你舊的，可以運作的情形·

3 暫存專用名稱伺服器

DNS 配置的首種嘗試，對撥接使用者非常有用·

一台暫存專用名稱伺服器將會爲名稱查詢找出答案並且在下一次你需要那個名稱的時候記得答案·

首先你需要一個稱爲 `/etc/named.boot` 的檔案·當 `named` 啓動時會讀取這個檔案· 目前它應該單純地包含:

```
;  Boot file for nicolais caching name server
;
directory /var/named
;
; type          domain                          source file or host
cache           .                               root.cache
primary         0.0.127.in-addr.arpa            pz/127.0.0
```

非常重要: 在這文件的某些版本中這個檔案的這份列表會在第一個非空白字元前包含一些空格或 tab 鍵· 這些不應該出現在檔案中·如果你從這份 HOWTO 剪貼下來，記得刪除任何前面的空白·

‘`directory`’ 這一行告訴 `named` 到哪裡去找尋檔案·所有其後命名的檔案都將是相對於此目錄的· 根據 Linux 檔案系統標準正確的目錄應該是在 `/var/named`· 因此 `pz` 是位於 `/var/named` 之下的，也就是，`/var/named/pz`·

稱爲 /var/named/root.cache 的這個檔案是在此命名的． 這個 /var/named/root.cache 應該包含:

```
 .       518400  NS      D.ROOT-SERVERS.NET.
 .       518400  NS      E.ROOT-SERVERS.NET.
 .       518400  NS      I.ROOT-SERVERS.NET.
 .       518400  NS      F.ROOT-SERVERS.NET.
 .       518400  NS      G.ROOT-SERVERS.NET.
 .       518400  NS      A.ROOT-SERVERS.NET.
 .       518400  NS      H.ROOT-SERVERS.NET.
 .       518400  NS      B.ROOT-SERVERS.NET.
 .       518400  NS      C.ROOT-SERVERS.NET.
;
D.ROOT-SERVERS.NET.     3600000 A       128.8.10.90
E.ROOT-SERVERS.NET.     3600000 A       192.203.230.10
I.ROOT-SERVERS.NET.     3600000 A       192.36.148.17
F.ROOT-SERVERS.NET.     3600000 A       192.5.5.241
G.ROOT-SERVERS.NET.     3600000 A       192.112.36.4
A.ROOT-SERVERS.NET.     3600000 A       198.41.0.4
H.ROOT-SERVERS.NET.     3600000 A       128.63.2.53
B.ROOT-SERVERS.NET.     3600000 A       128.9.0.107
C.ROOT-SERVERS.NET.     3600000 A       192.33.4.12
```

記住我說過要去掉空白字元!

這個檔案描述了在這個世界上的根名稱伺服器．這會隨時間而改變並且必須加以維護． 如何能保持它隨時更新請參見6 (維護篇)． 在 named 的線上使用手冊裡有這個檔案的描述，但是這，以我看來，最適合那些已經了解 named 的人們．

在 named.boot 裡的下一行是 primary 這一行． 我將會在稍後的章節裡解釋它的用法，目前只要把它設爲在 pz 子目錄下一個稱爲 127.0.0 的檔案:

```
@               IN      SOA     linux.bogus. hostmaster.linux.bogus. (
                                1       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

接下來，你需要一份看起來像這樣的 /etc/resolv.conf 檔案:

```
search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1
```

‘search’ 這一行指出對於任何你想連往的主機名稱應該搜尋的領域． ‘nameserver’ 這一行指出你的機器可以在哪個位址上找到一台名稱伺服器， 在這個例子中是你自己的這台機器，因爲你在它上面執

行 named· 如果你想列出好幾個名稱伺服器把它們都放在一行 ‘nameserver’ 裡，用空格隔開· (注意: named 從不讀取這個檔案，而是使用 named 的名稱解答器會讀取·)

來說明一下這個檔案有什麼作用: 如果某個客户端嘗試要找尋 foo 的話， 那麼首先嘗試的是 foo.subdomain.your-domain.edu 這個名稱， 然後接下來是 foo.your-fomain.edu 這個名稱，最後則是 foo 這個名稱· 如果有某個客户端嘗試要找尋 sunsite.unc.edu 的話， 那首先嘗試的是 sunsite.unc.edu.subdomain.your-domain.edu 這個名稱(沒錯，它眞笨! 不過它就是這麼做)， 然後接下來是 sunsite.unc.edu.your-domain.edu 這個名稱，最後則會是 sunsite.unc.edu 這個名稱· 你可能不會想放太多領域到 search 該行裡去，搜尋它們會多花時間·

這個範例假設你屬於 subdomain.your-domain.edu 這個領域， 那麼你的機器，可能會稱爲 your-machine.subdomain.your-domain.edu· 在 search 這行裡不應該包含你的 TLD (頂層領域 Top Level Domain，在這個例子中是 edu 這個領域)· 如果你經常需要連線到在另外一個領域裡的主機你麼你可以把該領域像這樣地加進 search 這行裡:

```
search subdomain.your-domain.edu your-domain.edu other-domain.com
```

依此類推·很明顯的是你得放入眞實的領域名稱來取代這些名稱·請注意在領域名稱的最後面並沒有句號 ’.’·

接下來，根據你 libc 版本的不同需要修正 /etc/nsswitch.conf 或者是 /etc/host.conf 檔案·

/etc/nsswitch.conf

這是一個很長的檔案，它指出到何處去取得各種不同的資料型態，從什麼檔案或是資料庫取得· 它的頂端經常會包含一些有用的註解·找出以 ‘hosts:’ 作爲開頭的那一行，它應該是這樣:

```
hosts:       files dns
```

如果檔案裡沒有以 ‘hosts:’ 作爲開頭的行那麼把上面這一行加上去· 它是說程式應該先在 /etc/hosts 檔案裡找尋，然後根據 resolv.conf 詢問 DNS·

/etc/host.conf

它可能包含有數行，其中應該有一行以 order 作爲開始而且它看起來會像這樣:

```
order hosts,bind
```

如果檔案裡沒有 ‘order’ 這一行的話那麼你應該貼一份上去· 它告訴主機名稱解析函式先在 /etc/hosts 裡找尋，然後查問名稱伺服器 (在 resolv.conf 裡你說在 127.0.0.1 這個地方)· 在大部分 Linux 發行套件中最後這兩個檔案的文件在 resolv(8) 的線上使用手冊中(執行 ‘man 8 resolv’ 即可)· 這份線上用手冊我覺得可看，而且每個人，特別是 DNS 管理者，都應該要閱讀它· 現在就做，如果你對你自己說“我稍後將會去做”，你將永遠不會去接近它·

3.1 起始 named

這些全部完成後就可以起始 named 了·如果你使用撥接連線的話那麼請先連上網路· 鍵入 ‘ndc start’ 並且按下 RETURN 鍵，沒有選項· 如果它不行的話那麼試著使用 ‘/usr/sbin/ndc start’ 來取代· 再不行的話請參考 9 (FAQ) 一節· 現在你可以測試你的設定·當你在起始 named 的時候如果你觀察一下 (使用 tail -f /var/adm/messages 指令)系統記錄訊息檔案 (通常是稱爲 /var/adm/messages 的檔案，但也可能在 /var/log 下，或是叫 syslog 的檔案) 那麼你應該會看見像樣的一些東西:

```
Jun 30 21:50:55 roke named[2258]: starting.  named 4.9.4-REL Sun Jun 30 21:29:03 MET DST 1996   janl@roke.slip
Jun 30 21:50:55 roke named[2258]: cache zone "" loaded (serial 0)
Jun 30 21:50:55 roke named[2258]: primary zone "0.0.127.in-addr.arpa" loaded (serial 1)
```

如果有任何關於錯誤的訊息那麼就是有個錯誤發生，named 將會指名有錯誤的檔案(我想是 named.boot 以及 root.cache 其中之一:-)· 殺掉 named 程序並回頭檢查那些檔案·

現在可以用 nslookup 來檢查一下你的工作:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>
```

如果這是你所得到的回應那麼它已經能夠運作·我們希望是這樣·得到任何其它回應都請回頭檢查每一件事· 每一次你改變 named.boot 檔案之後你都得使用 ndc restart 這個指令重新起始 named 程式·

現在你可以輸入查詢·嘗試找尋某些靠近你的機器·pat.uio.no 離我不遠，在 Oslo 的大學裡:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Name:    pat.uio.no
Address:  129.240.2.50
```

現在 nslookup 要求你的 named 找尋 pat.uio.no 這台機器· 然後它(named)聯繫在你 root.cache 檔案裡所指名的名稱伺服機器其中一台， 並且從那裡查問它該如何繼續下去·在你取得結果之前可能得花費一點時間， 因為它搜尋你在 /etc/resolv.conf 裡指名的所有領域·

如果你再試一次的話那麼你將會得到:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address:  129.240.2.50
```

注意這回我們所得到的 ‘Non-authoritative answer:’ 這一行· 這代表 named 此次並未到網路外去查問，取而代之的是在它的暫存區裡找尋並且在那裡找到答案· 但是暫存的資訊可能會過時·所以它藉由 ‘Non-authorative answer:’ 來知會你有這個(很輕微的)危險性存在· 當 nslookup 說這是你第二次查問某台主機時，這是 named 能暫存該項資訊並且正常運作的一個訊息· 你可以使用 ‘exit’ 指令離開 nslookup 程式·

如果你是個撥接(ppp, slip) 使用者請閱讀8 (撥接連線專節)，裡面有一些給你的建議·

現在你知道如何設立一個能夠暫存的 named 系統·來杯啤酒，牛奶，或是任何你喜歡的東西來慶祝吧·

4 一個簡單的領域

如何建立你自己的領域

4.1 先來一點純理論

在我們眞的開始進行這一節以前我將會提供你一些關於 DNS 如何運作的理論· 而你應該閱讀它因爲這對你有益·如果你不‘想要’那麼至少你也得很快地略讀一下· 當你看到應該放進 `named.boot` 檔案裡去的內容時再停止這種略讀方式·

DNS 是一個階層式的系統·其頂端寫作‘.’而其發音爲‘root’· 在 . 之下有幾個頂層領域(TLDs)，最知名的是 ORG, COM, EDU 以及 NET 這幾個，但是還有更多·

在尋找一台機器名稱時查詢會以遞迴方法從頂端開始· 當你想要找出 `prep.ai.mit.edu` 的位址時你的名稱伺服器必須找到負責 `edu` 的一台名稱伺服器· 這個問題它會去查問 `root.cache` 檔案，而 . 伺服器會給它一份 `edu` 伺服器列表·

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

開始查問某台根伺服器·

```
> server c.root-servers.net.
Default Server:  c.root-servers.net
Address:  192.33.4.12
```

設定查詢型態爲 NS (名稱伺服器記錄 name server records)·

```
> set q=ns
```

查問關於 `edu.` 的資料·

```
> edu.
```

結尾的 . 在這裡非常重要，它告訴該伺服器我們所查問的 `edu` 是在 . 之下的那一個， 這稍能縮小搜尋的範圍·

```
edu     nameserver = A.ROOT-SERVERS.NET
edu     nameserver = H.ROOT-SERVERS.NET
edu     nameserver = B.ROOT-SERVERS.NET
edu     nameserver = C.ROOT-SERVERS.NET
edu     nameserver = D.ROOT-SERVERS.NET
edu     nameserver = E.ROOT-SERVERS.NET
edu     nameserver = I.ROOT-SERVERS.NET
edu     nameserver = F.ROOT-SERVERS.NET
edu     nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
B.ROOT-SERVERS.NET      internet address = 128.9.0.107
```

```
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
F.ROOT-SERVERS.NET      internet address = 192.5.5.241
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
```

這告訴我們 `*.root-servers.net` 服務 `edu.` 領域，所以我們可以藉此繼續查問 c 伺服器． 現在我們想要知道是誰負責下一層 `mit.edu.` 的領域名稱:

```
> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12

Non-authoritative answer:
mit.edu nameserver = STRAWB.mit.edu
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu

Authoritative answers can be found from:
STRAWB.mit.edu  internet address = 18.71.0.151
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
```

steawb, w20ns 以及 `bitsy` 負責 `mit` 領域，選擇其中一個並且查詢 `ai.mit.edu`:

```
> server W20NS.mit.edu.
```

主機名稱不分大小寫，但是我使用滑鼠來剪貼所以這些資料是螢幕的拷貝．

```
Server:  W20NS.mit.edu
Address:  18.70.0.160

> ai.mit.edu.
Server:  W20NS.mit.edu
Address:  18.70.0.160

Non-authoritative answer:
ai.mit.edu      nameserver = WHEATIES.AI.MIT.EDU
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = WHEATIES.AI.MIT.EDU
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
WHEATIES.AI.MIT.EDU     internet address = 128.52.32.13
```

```
WHEATIES.AI.MIT.EDU     internet address = 128.52.35.13
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.32.5
ALPHA-BITS.AI.MIT.EDU   internet address = 128.52.37.5
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.32.4
GRAPE-NUTS.AI.MIT.EDU   internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.32.6
TRIX.AI.MIT.EDU internet address = 128.52.38.6
MUESLI.AI.MIT.EDU       internet address = 128.52.32.7
MUESLI.AI.MIT.EDU       internet address = 128.52.39.7
```

所以 `weaties.ai.mit.edu` 是 `ai.mit.edu` 的一台名稱伺服器:

```
> server WHEATIES.AI.MIT.EDU.
Default Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13
```

現在我改變查詢的型態，我們已經找到該名稱伺服器所以現在我們將要詢問 `wheaties` 關於 `prep.ai.mit.edu` 它所知道的任何事情·

```
> set q=any
> prep.ai.mit.edu.
Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
        inet address = 18.159.0.42, protocol = tcp
         #21 #23 #25 #79
prep.ai.mit.edu preference = 1, mail exchanger = life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = wheaties.ai.mit.edu
ai.mit.edu      nameserver = grape-nuts.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
life.ai.mit.edu internet address = 128.52.32.80
alpha-bits.ai.mit.edu   internet address = 128.52.32.5
wheaties.ai.mit.edu     internet address = 128.52.35.13
wheaties.ai.mit.edu     internet address = 128.52.32.13
grape-nuts.ai.mit.edu   internet address = 128.52.36.4
grape-nuts.ai.mit.edu   internet address = 128.52.32.4
mini-wheats.ai.mit.edu  internet address = 128.52.32.11
mini-wheats.ai.mit.edu  internet address = 128.52.54.11
mintaka.lcs.mit.edu     internet address = 18.26.0.36
```

所以我們從 . 開始連續找出在領域名稱裡的下一層名稱伺服器· 如果你使用你自己的 DNS 伺服器而不是所有這些個其它的伺服器，你的 `named` 當然會暫存所有這些在爲你尋找這個答案時所找到的資訊，而且在一段時間內它不必再次查問·

一個比較起來很少被論及，但是同樣重要的是 `in-addr.arpa` 領域．它也像‘正常的’領域一樣是巢狀的．`in-addr.arpa` 讓我們可以在擁有主機位址的時候得知該主機的名稱．在這裡有件重要的事情要注意的是在 `in-addr.arpa` 這個領域中 ip 數字是以反向順序書寫的．如果你有某台機器的位址: 192.128.52.43 那麼 named 會以類似 `prep.ai.mit.edu` 這個範例的方式來處理: 找出 `arpa.` 的伺服器，找出 `in-addr.arpa.` 的伺服器，然後再找出 `192.in-addr.arpa.` 的伺服器，找出 `128.192.in-addr.arpa.` 的伺服器，接著找出 `52.128.192.in-addr.arpa.` 的伺服器，最後再找出所需之 `43.52.128.192.in-addr.arpa.` 的記錄．聰明乎? (說‘是的’)．頭兩年這反向的數字也引起過一些困擾．

其實我是騙你的．DNS 並非完完全全地像我告訴你的這樣運作．但是這已經夠接近的了．

4.2 我們自己的領域

現在來定義我們自己的領域．我們將會創造出 *linux.bogus* 這個領域並且定義其中的機器．我使用一個完全是虛擬出來的領域名稱以便確定我們不會擾亂到網路上的其它地方．

我們早就已經以 `named.boot` 裡的這一行開始了這個部份的設定:

```
primary         0.0.127.in-addr.arpa            pz/127.0.0
```

請注意在這個檔案裡的領域名稱結尾並沒有加上‘.’符號．第一行把定義 `0.0.127.in-addr.arpa` 的檔案命名爲 `pz/127.0.0`．我們早已經建立了這個檔案，它是這樣的:

```
@               IN      SOA     linux.bogus. hostmaster.linux.bogus. (
                                1       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      ns.linux.bogus.
1                       PTR     localhost.
```

請注意在這個檔案裡所有的完整領域名稱結尾的‘.’符號，這與上面提到的 named.boot 檔案形成對比．有些人喜歡以 `$ORIGIN` 指令啓始每個區域檔案，但這是不必要的．一個區域檔案的基點(就是其所屬的 DNS 階層架構位置)是在 `named.boot` 檔案的‘領域’行裡指定的，在這個例子裡是 `0.0.127.in-addr.arpa`．

這個‘區域檔案’中包含三種‘資源記錄’(resource records, RRs): 一個是 SOA 資源記錄．一個是 NS 資源記錄以及一個 PTR 記錄．SOA 是授權起始(Start Of Authority)的縮寫．‘@’是個意思爲基點的特殊標記，而因爲這個檔案的‘領域’行說是 `0.0.127.in-addr.arpa` 所以第一行實際上是指

```
0.0.127.IN-ADDR.ARPA. IN      SOA ...
```

NS 是名稱伺服器資源記錄，它告訴 DNS 什麼機器是這個領域 `0.0.127.in-addr.arpa` 的名稱伺服器，也就是 `ns.linux.bogus.` 而最後的 PTR 記錄說 1 (等於是 `1.0.0.127.IN-ADDR.ARPA`, 也就是 `127.0.0.1`)的名稱是 localhost．

SOA 這個記錄是所有區域檔案的序文，而且在每一個區域檔案裡都應該有這唯一的一個，最開頭的記錄．它描述該區域，它從何而來(一台稱爲 `linux.bogus` 的機器)，誰負責其內容(`hostmaster@linux.bogus`)

，這個區域檔案是什麼版本(serial: 1)，以及其它必須做的，有關暫存與次要名稱伺服器的事．剩下的欄位如 refresh, retry, expire 以及 minimum 你可以使用這份文件裡所用的數字而且應該不會出問題．

現在重新起始你的 `named`(使用 `ndc restart` 指令)並使用 `nslookup` 來檢驗我們做了什麼:

```
$ nslookup

Default Server:  localhost
Address:  127.0.0.1

> 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1
```

所以它管理從 `127.0.0.1` 得到 `localhost` 的過程，很好．現在開始我們的主要任務，`linux.bogus` 這個領域，在 `named.boot` 裡插入新的一行 `primary` 指令:

```
primary               linux.bogus                     pz/linux.bogus
```

注意，在 `named.boot` 檔案裡領域名稱的結尾還是沒有 '.' 符號．

在這個 `linux.bogus` 區域檔案裡我們將會放入一些完全虛擬的資料:

```
;
; Zone file for linux.bogus
;
; Mandatory minimum for a working domain
;
@       IN      SOA     linux.bogus. hostmaster.linux.bogus. (
                        199511301       ; serial, todays date + todays serial #
                        28800           ; refresh, seconds
                        7200            ; retry, seconds
                        3600000         ; expire, seconds
                        86400 )         ; minimum, seconds
                NS      ns.linux.bogus.
                NS      ns.friend.bogus.
                MX      10 mail.linux.bogus   ; Primary Mail Exchanger
                MX      20 mail.friend.bogus. ; Secondary Mail Exchanger

localhost       A       127.0.0.1
ns              A       127.0.0.2
mail            A       127.0.0.4
```

關於 SOA 記錄有兩件事需要注意．首先 `ns.linux.bogus` 必須是一台具有 A 記錄的真正機器．在 SOA 記錄中用 CNAME 記錄爲名稱的機器是不合法的．它的名字不一定要是 `ns`，它可以是任何合法的主機名稱．再來，`hostmaster.linux.bogus` 應該被視爲 `hostmaster@linux.bogus`，這應該是一個

郵件位址或別名，是維護這個 DNS 的人經常讀信的位址·任何關於此領域的信件會被送到這個位址·它的名字不一定要是 `hostmaster`，它可以是任何合法的電子郵件位址，然而這個位址 ‘`hostmaster`’ 應該能工作正常·

這個檔案裡有一種新的資源記錄型態，即 MX 型態，或是郵件交換者資源記錄(Mail eXchanger RR)·這種資源記錄型態告訴郵遞系統地址 `someone@linux.bogus` 的郵件要寄送到哪裡，換句話說也就是應該寄送到 `mail.linux.bogus` 或是 `mail.friend.bogus`· 在每個機器名稱前面的數字是 MX 資源記錄的優先權，數字比較低 (10) 的資源記錄是郵件主要應該寄往的機器·如果失敗可以把它寄往數字比較高的機器，一台次要的郵件處理者，也就是在這裡具有優先權 20 的 `mail.friend.bogus`·

用 `ndc restart` 重新起始 `named`·以 `nslookup` 檢驗結果:

```
$ nslookup
> set q=any
> linux.bogus
Server:  localhost
Address:  127.0.0.1

linux.bogus
        origin = linux.bogus
        mail addr = hostmaster.linux.bogus
        serial = 199511301
        refresh = 28800 (8 hours)
        retry   = 7200 (2 hours)
        expire  = 604800 (7 days)
        minimum ttl = 86400 (1 day)
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
linux.bogus     preference = 20, mail exchanger = mail.friend.bogus
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 127.0.0.2
mail.linux.bogus        internet address = 127.0.0.4
```

小心地檢驗你將會發現一個錯誤·這一行

```
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
```

全都錯了·它應該是

```
linux.bogus     preference = 10, mail exchanger = mail.linux.bogus
```

我故意犯了個錯誤所以你可以藉此學習:-) 仔細看看該區域檔案我們會發現這一行

```
@               MX      10 mail.linux.bogus     ; Primary Mail Exchanger
```

遺漏了一個句點·或是說多了個 ‘`linux.bogus`’· 在區域檔案裡如果一個機器名稱不是以句點結尾那麼會在其結尾加入基點·所以不論是

```
@               MX      10 mail.linux.bogus.    ; Primary Mail Exchanger
```

或是

```
@               MX      10 mail                 ; Primary Mail Exchanger
```

都是正確的．我比較喜歡後面這種形式，它需要的打字比較少． 在一個區域檔案裡領域名稱應該要不就是寫出來並以‘.’結尾或者就是一點都不要包含進去， 而在這種情況下其領域預設爲基點．其它人也許喜歡採用另一種方式．

我必須強調的是在 `named.boot` 檔案裡領域名稱後面不應該有‘.’的存在． 你不知道有多少次因爲多了或少了一個‘.’而搞砸了並且對許多人造成困擾．

所以在加入我強調的重點後得到一份新的區域檔案，其中還包含一些額外的資訊:

```
;
; Zone file for linux.bogus
;
; Mandatory minimum for a working domain
;
@       IN      SOA     linux.bogus.    hostmaster.linux.bogus. (
                        199511301       ; serial, todays date + todays serial #
                        28800           ; refresh, seconds
                        7200            ; retry, seconds
                        604800          ; expire, seconds
                        86400 )         ; minimum, seconds

                NS      ns              ; Inet Address of name server
                NS      ns.friend.bogus.
                MX      10 mail         ; Primary Mail Exchanger
                MX      20 mail.friend.bogus. ; Secondary Mail Exchanger

localhost       A       127.0.0.1
ns              A       127.0.0.2
mail            A       127.0.0.4
;
; Extras
;
@               TXT     "Linux.Bogus, your DNS consultants"

ns              MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "Pentium" "Linux 1.2"
                TXT     "RMS"
richard         CNAME   ns
www             CNAME   ns

donald          A       127.0.0.3
                MX      10 mail
```

```
                MX      20 mail.friend.bogus.
                HINFO   "i486"  "Linux 1.2"
                TXT     "DEK"

mail            MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "386sx" "Linux 1.0.9"

ftp             A       127.0.0.5
                MX      10 mail
                MX      20 mail.friend.bogus.
                HINFO   "P6" "Linux 1.3.59"
```

你也許會想要移動前面三個 A 記錄以便讓它們靠近其它的相關記錄，而不是像這樣放在最前端·

這裡有幾個新的資源記錄: 主機資訊(HINFO: Host INFOrmation) 包括兩個部份， 使用引號涵括每個部份是個好習慣·第一個部份是機器上的硬體或是中央處理單元， 而第二個部份是機器上的軟體或是作業系統·`ns` 有一顆 Pentium CPU 並且執行 Linux 1.2 系統· TXT 記錄是個隨意的文字記錄，你可以用它來記錄任何事情· 正式名稱(CNAME: Canonical NAME)可以用來給每台機器數個名稱· 所以 `richard` 以及 `www` 都是 `ns` 的一個別名· 很重要的一點是 A, MX, CNAME, 以及 SOA 記錄永遠不該參照 CNAME 記錄設定的別名， 它們只應該參照 A 記錄所設定的名稱，所以像這樣的記錄是錯的

```
foobar  CNAME   richard                 ; NO!
```

不過這樣是對的

```
foobar  CNAME   ns                      ; Yes!
```

還有一點也很重要的是注意正式名稱所設定的對電子郵遞位址而言不是合法主機名稱: `webmaster@www.linux.bogus` 以上面的設定而言是一個不合法的電子郵遞位址· 即使它在你的系統上可以運作，可以預料的是很少有電子郵件管理者會去實行這項規則· 避免這個問題的方法是使用 A 記錄(或者也可能是一些其它的，像是 MX 記錄)來取代之:

```
www     A       127.0.0.2
```

Paul Vixie, 主要的 `named` 專家，建議不要使用 CNAME 這個設定· 所以應該考慮不要很認眞地去使用它·

藉由執行 `ndc reload` 載入新的資料庫，這會使 `named` 再一次讀取其檔案·

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

> ls -d linux.bogus
```

這意指應該列出所有的記錄

```
[localhost]
 linux.bogus.                 SOA   ns.linux.bogus hostmaster.linux.bogus. (199511301 28800 7200 604800 8640
 linux.bogus.                 NS    ns.linux.bogus
 linux.bogus.                 NS    ns.friend.bogus
 linux.bogus.                 MX    10   mail.linux.bogus
 linux.bogus.                 MX    20   mail.friend.bogus
 linux.bogus.                 TXT   "Linux.Bogus, your DNS consultants"
 localhost                    A     127.0.0.1
 mail                         A     127.0.0.4
 mail                         MX    10   mail.linux.bogus
 mail                         MX    20   mail.friend.bogus
 mail                         HINFO 386sx        Linux 1.0.9
 donald                       A     127.0.0.3
 donald                       MX    10   mail.linux.bogus
 donald                       MX    20   mail.friend.bogus
 donald                       HINFO i486         Linux 1.2
 donald                       TXT   "DEK"
 www                          CNAME ns.linux.bogus
 richard                      CNAME ns.linux.bogus
 ftp                          A     127.0.0.5
 ftp                          MX    10   mail.linux.bogus
 ftp                          MX    20   mail.friend.bogus
 ftp                          HINFO P6           Linux 1.3.59
 ns                           A     127.0.0.2
 ns                           MX    10   mail.linux.bogus
 ns                           MX    20   mail.friend.bogus
 ns                           HINFO Pentium      Linux 1.2
 ns                           TXT   "RMS"
 linux.bogus.                 SOA   ns.linux.bogus hostmaster.linux.bogus. (199511301 28800 7200 604800 8640
```

很好．讓我們檢查它對於單獨的 `www` 會說什麼:

```
> set q=any
> www.linux.bogus.
Server:  localhost
Address:  127.0.0.1

www.linux.bogus canonical name = ns.linux.bogus
```

...換句話說，`www.linux.bogus` 眞正的名稱是 `ns.linux.bogus`．

```
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 127.0.0.2
```

而 `ns.linux.bogus` 擁有 127.0.0.2 這個位址．看起來也很好．

4.3 繼續深入

當然，這個領域是高度虛擬的，裡面所有的位址也都是，而這或許會讓你有些混淆． 眞實領域的一個眞實例子請見下一節．

5 眞實領域的例子

這裡我們列出一些眞正的區域檔案

用者建議我加上一個可以運作的眞實例子，因爲我對眞正可運作的領域及虛擬範例之間差別解釋的不夠清楚．

關於這個例子有件事必須聲明: 千萬不要把它放到你的名稱伺服器上! 把它當作參考資料來閱讀就好．如果你想做實驗，用上面虛擬的例子． 我經過 LAND-5 的 David Bullock 的同意才使用這個例子．這些檔案的日期是 1996 年九月二十日， 因此如果你現在查詢 LAND-5 的名稱伺服器可能會發現有所不同． 還有，請記住，刪去前面的空白 :-)

5.1 /etc/named.boot (或 /var/named/named.boot)

我們發現需要兩行 `primary` 的反向區域: 一個是 `127.0.0`，另一個是 LAND-5 的子網路 `206.6.177`．還有一行給 `land-5.com` 的正向區域．還有一點要注意的是不像我在這篇文章裡將檔案放在 `pz` 目錄裡， 他將它們放到叫做 `zone` 的目錄．

```
;  Boot file for LAND-5 name server
;
directory /var/named
;
; type          domain                          source file or host
cache           .                               root.cache
primary         0.0.127.in-addr.arpa            zone/127.0.0
primary         177.6.206.in-addr.arpa          zone/206.6.177
primary         land-5.com                      zone/land-5.com
```

5.2 /var/named/root.cache

記住這個檔案是動態的，這邊列出來的是舊的．你最好自己用 `dig` 產生一個新的．

```
; <<>> DiG 2.1 <<>>
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr rd ra; Ques: 1, Ans: 9, Auth: 0, Addit: 9
;; QUESTIONS:
;;      ., type = NS, class = IN

;; ANSWERS:
 .       518357  NS      H.ROOT-SERVERS.NET.
 .       518357  NS      B.ROOT-SERVERS.NET.
 .       518357  NS      C.ROOT-SERVERS.NET.
 .       518357  NS      D.ROOT-SERVERS.NET.
 .       518357  NS      E.ROOT-SERVERS.NET.
 .       518357  NS      I.ROOT-SERVERS.NET.
```

```
 .      518357  NS      F.ROOT-SERVERS.NET.
 .      518357  NS      G.ROOT-SERVERS.NET.
 .      518357  NS      A.ROOT-SERVERS.NET.

;; ADDITIONAL RECORDS:
H.ROOT-SERVERS.NET.     165593  A       128.63.2.53
B.ROOT-SERVERS.NET.     165593  A       128.9.0.107
C.ROOT-SERVERS.NET.     222766  A       192.33.4.12
D.ROOT-SERVERS.NET.     165593  A       128.8.10.90
E.ROOT-SERVERS.NET.     165593  A       192.203.230.10
I.ROOT-SERVERS.NET.     165593  A       192.36.148.17
F.ROOT-SERVERS.NET.     299616  A       192.5.5.241
G.ROOT-SERVERS.NET.     165593  A       192.112.36.4
A.ROOT-SERVERS.NET.     165593  A       198.41.0.4

;; Total query time: 250 msec
;; FROM: land-5 to SERVER: default ---- 127.0.0.1
;; WHEN: Fri Sep 20 10:11:22 1996
;; MSG SIZE  sent: 17  rcvd: 312
```

5.3 /var/named/zone/127.0.0

不過是基本的，不能省略的 SOA 記錄，以及將 `127.0.0.1` 指向 `localhost` 的記錄· 兩者都是必須的·不應有其它的東西在此檔案中·這個檔案可能永遠不必更新，除非你的名稱伺服器或管理人地址改變·

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609203       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      land-5.com.

1                       PTR     localhost.
```

5.4 /var/named/zone/land-5.com

在這裡我們看到這個必須的 SOA 記錄，必要的 NS 記錄· 我們可以看到它有一個次要的名稱伺服器於 `ns2.psi.net`· 這也應該是必要的，總是保有一個次要的伺服器做爲備份· 我們還看到做爲 LAND-5 負責所有不同服務的主機，它以許多的 CNAME 記錄做到這點 (另一個做法是用 A 記錄)·

就像你從 SOA 記錄看到的一樣，區域檔案以 `land-5.com` 開始，管理人是 `root@land-5.com`· 序號 (serial)以 yyyymmdd 的格式加上當天的號碼; 它可能是 1996 年九月二十號那天的第六個版本的區域檔案· 請記住序號必須單調地遞增，這裡它只用一位數字表示當天的號碼， 因此在編輯九次之後他必須等到明天才能再編輯這個檔案·[譯註] 因此應該考慮用兩位數字·

]

讀者可能會奇怪爲什麼不能用 1996092010 ？ 這個數字也是遞增的，所以當然可以· 但到了明天第一次編檔案時該用什麼呢? 如果用 199609211，那麼這個數字就比 1996092010 要小了· 這樣會有麻煩·因此作者建議應用兩位數字，如 1996092006·

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        10800           ; refresh, seconds
                        7200            ; retry, seconds
                        10800           ; expire, seconds
                        86400 )         ; minimum, seconds
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Primary Mail Exchanger

localhost       A       127.0.0.1

router          A       206.6.177.1

land-5.com.     A       206.6.177.2
ns              CNAME   land-5.com.
ftp             CNAME   land-5.com.
www             CNAME   land-5.com.
mail            CNAME   land-5.com.
news            CNAME   land-5.com.

funn            A       206.6.177.3
illusions       CNAME   funn.land-5.com.
@               TXT     "LAND-5 Corporation"

;
;       Workstations
;
ws_177200       A       206.6.177.200
                MX      10 land-5.com.   ; Primary Mail Host
ws_177201       A       206.6.177.201
                MX      10 land-5.com.   ; Primary Mail Host
ws_177202       A       206.6.177.202
                MX      10 land-5.com.   ; Primary Mail Host
ws_177203       A       206.6.177.203
                MX      10 land-5.com.   ; Primary Mail Host
ws_177204       A       206.6.177.204
                MX      10 land-5.com.   ; Primary Mail Host
ws_177205       A       206.6.177.205
                MX      10 land-5.com.   ; Primary Mail Host
; {Many repetitive definitions deleted - SNIP}
ws_177250       A       206.6.177.250
```

```
                MX      10 land-5.com.    ; Primary Mail Host
ws_177251       A       206.6.177.251
                MX      10 land-5.com.    ; Primary Mail Host
ws_177252       A       206.6.177.252
                MX      10 land-5.com.    ; Primary Mail Host
ws_177253       A       206.6.177.253
                MX      10 land-5.com.    ; Primary Mail Host
ws_177254       A       206.6.177.254
                MX      10 land-5.com.    ; Primary Mail Host
```

另一件值得注意的事所有的工作站都沒有個別的名稱，而是用一開頭加上最後兩位的 IP 數字· 用這樣的慣例可以大量地簡化維護工作，但可能有點不方便，而且，事實上可能是你的客户不滿的來源·

5.5 /var/named/zone/206.6.177

我將在後面說明這個檔案·

```
@               IN      SOA     land-5.com. root.land-5.com. (
                                199609206       ; Serial
                                28800   ; Refresh
                                7200    ; Retry
                                604800  ; Expire
                                86400)  ; Minimum TTL
                        NS      land-5.com.
                        NS      ns2.psi.net.
;
;       Servers
;
1       PTR     router.land-5.com.
2       PTR     land-5.com.
3       PTR     funn.land-5.com.
;
;       Workstations
;
200     PTR     ws_177200.land-5.com.
201     PTR     ws_177201.land-5.com.
202     PTR     ws_177202.land-5.com.
203     PTR     ws_177203.land-5.com.
204     PTR     ws_177204.land-5.com.
205     PTR     ws_177205.land-5.com.
; {Many repetitive definitions deleted - SNIP}
250     PTR     ws_177250.land-5.com.
251     PTR     ws_177251.land-5.com.
252     PTR     ws_177252.land-5.com.
253     PTR     ws_177253.land-5.com.
254     PTR     ws_177254.land-5.com.
```

反向區域似乎是整個設定中最讓人頭痛的部份·它是在你有機器的 IP 數字時查詢機器名稱用的· 例如: 你有一部 irc 伺服器接受 irc 客户端的連線· 但你是挪威的伺服器因此你只希望接受來自挪威及其它斯堪地那維亞國家的連線· 當你接到來自客户端的連線時 C 程式庫能夠告訴你連線機器的 IP 位址，因爲客户端的 IP 數字是包含在傳入的網路封包中的· 然後你可以呼叫一函式 `gethostbyaddr` 以給定的 IP 查詢機器名稱· `gethostbyaddr` 函式會向 DNS 詢問，DNS 就會出去查詢這台機器名稱· 假設客户端來自 `ws_177200.land-5.com`，C 程式庫提供給 irc 伺服器的 IP 是 `206.6.177.200`· 爲找出機器的名稱我們要查詢 `200.177.6.206.in-addr.arpa`·DNS 伺服器首先找到負責 `arpa.` 的伺服器· 然後找到 `in-addr.arpa.` 的伺服器，再來是反過來的 `206`，然後是 `6`， 最後找到在 LAND-5 負責 `177.6.206.in-addr.arpa` 區域的伺服器· 由此我們得到 `200.177.6.206.in-addr.arpa` 的答案是 'PTR `ws_177200.land-5.com`' 記錄， 意指擁有 IP `206.6.177.200` 的名稱是 `ws_177200.land-5.com`· 不過就像在解釋 `prep.ai.mit.edu` 如何查詢時所說，這過程稍微有點不對·

回到 irc 伺服器的例子·irc 伺服器只接再來自斯堪地那維亞國家的連線，也就是， *.no, *.se, *.dk· `ws_177200.land-5.com` 這個名字顯然並不符合任何這些· 因此伺服器會拒絕連線·如果沒有經由 `in-addr.arpa` 區域到 `206.2.177.200` 的反向對應， 伺服器將無法查到機器名稱而將會拿 `206.2.177.200` 來和 *.no, *.se 及 *.dk 比較，當然找不到符合的·

有些人會告訴你反向對應查詢只對伺服器重要，或是一點也不重要· 當然不是: 許多 ftp, news, irc 甚至 http 伺服器將不接受無法查到名稱的機器連線· 因此事實上機器的反向名稱對應是必須的·

6 維護工作

維持它的運作·

維持它們的繼續執行之外，對於 `named` 你還有個維護的任務得要做· 那就是維持 `root.cache` 檔案的更新·最簡單的方法是使用 `dig` 程式， 首先不加任何參數執行 `dig` 程式，你將會取得從你自己伺服器得到的 `root.cache`· 然後以 `dig @rootserver . ns` 查問所列出的根伺服器其中之一· 你將會注意到這份輸出看起來非常地像一個 `root.cache` 檔案，除了一堆額外的數字以外·這些數字不會有什麼妨礙· 把它存放到檔案裡(`dig @e.root-servers.net . ns` >`root.cache.new`)並且用它來取代原本舊的 `root.cache` 檔案·

取代了原先的檔案之後要記得重新啓動 `named` 程式·

Al Longyear 寄給我這個指令稿，它可以自動執行來更新 `named.cache`，爲它安裝個 crontab 項目然後忘了它· 這個指令稿假設你的電子郵件可以運作而且 ‘hostmaster’ 這個郵件別名有定義·你應該修訂它以便符合你的設定·

```
#!/bin/sh
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
(
 echo "To: hostmaster <hostmaster>"
 echo "From: system <root>"
 echo "Subject: Automatic update of the named.boot file"
 echo
```

```
 export PATH=/sbin:/usr/sbin:/bin:/usr/bin:
 cd /var/named

 dig @rs.internic.net . ns >root.cache.new

 echo "The named.boot file has been updated to contain the following information:"
 echo
 cat root.cache.new

 chown root.root root.cache.new
 chmod 444 root.cache.new
 rm -f root.cache.old
 mv root.cache root.cache.old
 mv root.cache.new root.cache
 ndc restart
 echo
 echo "The nameserver has been restarted to ensure that the update is complete."
 echo "The previous root.cache file is now called /var/named/root.cache.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

有些人可能用 ftp 從 Internic 得到 `root.cache` 檔案． 請不要用 ftp 來更新 `root.cache`，上面提到的方法更爲便利．

7 Bind version 8

Bind version 8 是最新的 bind 技術．我還沒有嘗試使用，但 David E. Smith <*dave@bureau42.ml.org*> 有．他寫了這小節的其餘部份．

沒有太多的不同．除了使用 `named.conf` 而不是 `named.boot`, 其餘都相同． 而且 bind 8 包含了一個 perl 指令稿來將舊格式的檔案轉換成新的． 例如一份暫存專用名稱伺服器的 `named.boot` (舊格式):

```
directory /var/named
cache   .                                               root.hint
primary 0.0.127.IN-ADDR.ARPA                            127.0.0.zone
primary localhost                                       localhost.zone
```

在指令列下，於 `bind8/src/bin/named` 目錄中鍵入:

```
named-bootconf.pl < named.boot > named.conf
```

這會產生 `named.conf`:

```
// generated by named-bootconf.pl

options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hint";
};

zone "0.0.127.IN-ADDR.ARPA" {
        type master;
        file "127.0.0.zone";
};

zone "localhost" {
        type master;
        file "localhost.zone";
};
```

它將 `named.boot` 中的每樣東西都轉換過來，然而它並未加入所有 bind 8 所允許的新強化及配置選項· 這裡有份較完整的 `named.conf` 做同樣的事，但是效率更好一點·

```
// This is a configuration file for named (from BIND 8.1 or later).
// It would normally be installed as /etc/named.conf.
// The only change made from the ‘stock’ named.conf (aside from this
// comment :) is that the directory line was uncommented, since I
// already had the zone files in /var/named.

options {
        directory "/var/named";
        check-names master warn;                /* default. */
        datasize 20M;
};

zone "localhost" IN {
        type master;
        file "localhost.zone";
        check-names fail;
        allow-update { none; };
        allow-transfer { any; };
};

zone "0.0.127.in-addr.arpa" IN {
        type master;
        file "127.0.0.zone";
```

```
        check-names fail;
        allow-update { none; };
        allow-transfer { any; };
};

zone "." IN {
        type hint;
        file "root.hint";
};
```

`bind8/src/bin/named/test` 裡有這個範例，以及領域檔案的拷貝，許多人可以直接拿來修改使用・

領域檔案及 `root.hint` (`root.cache`) 完全相同，更新它們的指令也是・

8 撥接連線的自動設定

這一節解釋我如何設定以便讓每個過程自動化・ 我的方法可能一點都不適合你，但是你可能從我所作的一些事情裡得到一些點子・ 同時，我使用 PPP 撥接，然而有許多人使用 slip 或是 cslip 連線方式，所以你的設定裡幾乎每個地方都可能跟我的不同・ 但是 slip 使用的 `dip` 程式應該能夠完成我所作的許多事情・

一般來說，當我沒有連上網路時我的 `resolv.conf` 檔案單純地包含這一行

```
domain uio.no
```

這確保我不必等待主機名稱解析函式庫去嘗試聯繫某台不可能幫助我的名稱伺服器・ 但是當我連上線的時候我想要起始我的 `named` 並且擁有一個看起來像前面所描述的 `resolv.conf` 檔案・ 我藉由保持兩份 `resolv.conf` 的‘樣板’檔案，`resolv.conf.local` 以及 `resolv.conf.connected` 來解決問題・ 後面這一個看起來像前面所描述過的 `resolv.conf` 檔案・

要自動化連線到網路的過程我執行一個稱爲 ‘ppp-on’ 的指令稿:

```
#!/bin/sh
echo calling...
pppd
```

pppd 有個稱爲 `options` 的檔案，它告訴 pppd 如何取得連線的一些特殊事項・ 一旦我的 ppp 連線完成後 pppd 起始一個稱爲 `ip-up` 的指令(這在 pppd 的線上使用手冊裡有描述)・ 這裡是該指令稿裡面的一部份:

```
#!/bin/sh
interface="$1"
device="$2"
speed="$3"
myip="$4"
upip="$5"
 ...
```

```
cp -v /etc/resolv.conf.connected /etc/resolv.conf
 ...
/usr/sbin/named
```

換句話說，我在這裡起始我的 named 程式．當 ppp 離線時 pppd 執行一個稱爲 ip-down 的指令稿:

```
#!/bin/sh
cp /etc/resolv.conf.local /etc/resolv.conf
read namedpid < /var/run/named.pid
kill $namedpid
```

所以這在連線時配置並設定相關事宜並且在離線時解除該配置並結束相關程式．

某些程式，例如我想起 irc 與 talk，做了過多的假定，爲了讓其工作正常你必須修改你的 hosts 檔案． 我在 ip-up 上加入

```
cp /etc/hosts.ppp /etc/hosts
echo $myip      roke >>/etc/hosts
```

hosts.ppp 是簡單地包含了

```
127.0.0.1       localhost
```

而 echo 指令插入了我收到的 IP 號碼做爲我機器的名稱(roke)． 你應該改用你自己機器的名稱，可以用 hostname 指令查出．

當你沒有連線到網路上時執行 named 可能並不聰明， 這是因爲 named 將會嘗試送出查詢到網路上而且其終止時限(timeout)很長， 而每次有某些個程式嘗試解析一個名稱的時候你就得等待到這個終止時限． 如果你使用撥接的話你應該在連上網時起始 named 並且在離線時殺掉它． 不過請參考 9 (FAQ) 一節的提示．

某些人喜歡在慢速的連線上使用 forwarders 指令． 如果你的網際網路提供者在 1.2.3.4 以及 1.2.3.5 設有 DNS 伺服器那麼你可以插入這麼一行

```
forwarders 1.2.3.4 1.2.3.5
```

到 named.boot 檔案裡去． 這將會減低源自你主機的 IP 流量，可能稍微提昇速度．如果你是依線路的資料量付費的話這點特別重要． 這還有個附加價值，讓你脫離作爲一個暫存的 named 維護者所應負起的責任: 你不需要去更新一個空的 root.cache 檔案．

9 常見問題解答

本節中我將列出一些關於 DNS 以及本文件最常被提出的問題．還有解答 :-) 在寫信給我前請先閱讀過本節．

1. 如何在防火牆之內使用 DNS？

   一些提示: ‘forwarders’, ‘slave’, 以及參考本文件最後面的文獻列表．

2. 如何利用 DNS 提供的可用位址於某一服務，如 `www.busy.site`，以獲得負荷平衡或類似的效果?

   爲 `www.busy.site` 加上許多 **A** 記錄，而且使用 `bind 4.9.3` 或以後的版本． 這樣 `bind` 會使用 round-robin 演算法找尋答案．這不能在更早版本的 `bind` 上工作．

3. 我想在一封閉的 Intranet 上架設 DNS．我該怎麼做?

   把暫存檔案丟掉只留區域檔案．這也表示你將不需要更新暫存檔案．

4. 我的系統沒有 `ndc` 程式．我該怎麼辦?

   你的系統安裝過一舊的，可能過期的 `bind` 程式． 如果安全性對你很重要: 立刻將 `bind` 升級．如果不是，你也可以用這舊的． 你可直接執行 `named` 而不是 `ndc start`． `ndc reload` 變成 `named.reload` 而 `ndc restart` 變成 `named.restart`． 所有這些程式可能都在 `/usr/sbin` 裡．

5. 如何建立次要名稱伺服器?

   如果主要名稱伺服器的地址是 `127.0.0.1`，在你的次要名稱伺服器的 `named.boot` 檔案中放一行:

```
secondary      linux.bogus              127.0.0.1        sz/linux.bogus
```

6. 我希望離線時 `bind` 仍繼續執行．

   我收到來自 Ian Clark <*ic@deakin.edu.au*> 的信件解釋他如何辦到這點．

```
I run named on my 'Masquerading' machine here. I have
two root.cache files, one called root.cache.real which contains
the real root server names and the other called root.cache.fake
which contains...

--------------
; root.cache.fake
; this file contains no information
--------------

When I go off line I copy the root.cache.fake file to root.cache and
restart named.

When I go online I copy root.cache.real to root.cache and restart
named.

This is done from ip-down & ip-up respectively.

The first time I do a query off line on a domain name named doesn't
have details for it puts an entry like this in messages..

Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN

which I can live with.

It certainly seems to work for me. I can use the nameserver for
local machines while off the 'net without the timeout delay for
external domain names and I while on the 'net queries for external
domains work normally
```

7. 名稱伺服器將暫存資料保存在那兒? 有任何方法可以控制它的大小嗎?

   所有的暫存資料都保存在記憶體中，任何時刻它都不會被寫入磁碟中· 每次你殺掉`named`時這些暫存資料就沒有了·沒有任何方法可以控制暫存資料· `named`用一些簡單的法則來管理它·你不能用任何理由任何方法控制暫存資料或其大小· 如果你希望如此你可以自己去改`named`的程式·不過最好別這麼做·

8. 如何在重新起始`named`之間保存暫存資料? 我可以把它存下來嗎?

   不! `named`被殺掉時不會保留暫存資料· 這就是說暫存資料必須在每次你殺掉並重新起始`named`後重新建立· 沒有辦法讓`named`將暫存資料存到檔案裡· 如果你希望如此你可以自己去改`named`的程式·不過最好別這麼做·

10 如何成爲一個大型的DNS管理者

文件以及工具

存在眞正的文件·在線上的或是印好的·要跨出由小型DNS到大型DNS管理者的一步閱讀幾份這些文件是必要的· 印好的標準參考書是*DNS and BIND* by C. Liu and P. Albitz from O'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X· 我讀過，它很棒·在另一本*TCP/IP Network Administration*, by Craig Hunt from O'Reilly..., ISBN 0-937175-82-X有一節是關於DNS的·另一本對管理DNS很好的(或者是對任何這一類工作都很好)書則是*Zen and the Art of Motorcycle Maintenance*, 作者Robert M. Prisig書號ISBN 0688052304·

線上的資料你可以在這裡找到 `<http://www.dns.net/dnsrd/>`, `<http://www.isc.org/bind.html>`; 常見問題集，參考手冊(BOG; Bind Operations Guide)是協定的定義以及DNS研究的文獻· 這些文件我大部分沒有閱讀過，但是因此我不是個大型DNS管理者· 另一方面Arnt Gulbrandsen已經讀過BOG而且對此很熟稔 :-)· 網路新聞討論群*comp.protocols.tcp-ip.domains*是有關DNS的討論群· 此外還有好幾份關於DNS的RFCs，最重要的可能是這些:

RFC 2052

A. Gulbrandsen, P. Vixie, *A DNS RR for specifying the location of services (DNS SRV)*, October 1996

RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Address Allocation for Private Internets*, 02/29/1996.

RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 02/28/1996.

RFC 1713

A. Romao, *Tools for DNS debugging*, 11/03/1994.

RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.

RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.

RFC 1035

P. Mockapetris, *Domain names - implementation and specification*, 11/01/1987.

RFC 1034

P. Mockapetris, *Domain names - concepts and facilities*, 11/01/1987.

RFC 1033

M. Lottor, *Domain administrators operations guide*, 11/01/1987.

RFC 1032

M. Stahl, *Domain administrators guide*, 11/01/1987.

RFC 974

C. Partridge, *Mail routing and the domain system*, 01/01/1986.