

## 2.0 Einführung

Bei 30 verschiedenen Steinen im deutschen Scrabble (»A« bis »Z« plus Umlaute und Joker) und sieben Steinen auf einem Bänkchen können Sie Milliarden von unterschiedlichen Kombinationen ziehen. Und da die meisten der über 120.000 Wörter im Duden nach den Scrabble-Regeln erlaubt sind<sup>1</sup>, können Sie aus fast jeder Kombination eine ganze Menge Wörter bilden.

Bei DNS gibt es weniger als 300 mögliche Arten von Resource Records, und von diesen kann man nur eine Handvoll als gängig bezeichnen. Dennoch können Sie mit diesen Records eine bemerkenswerte Vielfalt interessanter Dinge anstellen.

Alle Resource Records haben in reinem Text geschrieben (so, wie sie in einer Zonendaten-Datei stehen) das folgende gemeinsame Format:

```
[owner] [TTL] [class] <type> <RDATA>
```

Die Felder in eckigen Klammern (»[« und »]«) sind optional, während die Felder in spitzen Klammern (»<« und »>«) unerlässlich sind. Rezept 2.1 erläutert, was passiert, wenn Sie eines oder mehrere dieser Felder weglassen.

Das RDATA-Feld besteht oft aus mehreren Unterfeldern. Die Anzahl der benötigten Unterfelder hängt vom Typ des Records ab. Zum Beispiel erfordern SOA-Records sieben RDATA-Unterfelder, während A- und NS-Records nur eins benötigen.

Eine Zonendaten-Datei enthält alle Resource Records, die mit Domain-Namen in einer Zone verbunden sind. Der primäre Master-Nameserver einer Zone lädt die Zonendaten-Datei und die Slave-Nameserver der Zone laden die Zonendaten vom primären Master.

---

<sup>1</sup> Anm. d. Ü.: Da Scrabble in Deutschland bei weitem nicht so populär ist wie in der englischsprachigen Welt, gibt es hier keinen Scrabble-Verband und demzufolge auch keine offizielle Wortliste.

## 2.1 Eine Zonendaten-Datei anlegen

### Problem

Sie müssen eine Datendatei für eine Zone anlegen.

### Lösung

Verwenden Sie Ihren Lieblingstexteditor, um eine Datei im Arbeitsverzeichnis des primären Master-Nameservers anzulegen. Benennen Sie diese Datei nach der Zone, deren Resource Records sie enthalten soll. Für die Zone *foo.example* könnten Sie die Zonendaten-Datei beispielsweise *db.foo.example* nennen.

Beginnen Sie die Datei mit einer *\$TTL*-Steueranweisung.<sup>2</sup> Diese teilt anderen (für diese Zone nicht autorisierten) Nameservern mit, wie lange sie Records aus dieser Zone im Cache halten können, legt also die standardmäßige *time to live* der Zone fest. Sie können den Wert als ganzzahlige Sekundenanzahl oder als Wert mit Maßeinheit angeben: eine ganze Zahl, auf die *s* für Sekunden, *m* für Minuten, *h* für Stunden, *d* für Tage oder *w* für Wochen folgt. Zum Beispiel können Sie einen TTL-Wert von einem Tag entweder so angeben:

```
$TTL 86400
```

oder so:

```
$TTL 1d
```

Sie können sogar zwei Werte mit Maßeinheit verknüpfen, und zwar folgendermaßen:

```
$TTL 1d12h
```

Üblich sind Time-to-live-Werte zwischen einer Stunde und einem Tag.

Fügen Sie als Nächstes einen SOA-Record für die Zone hinzu. Der SOA-Record enthält Informationen über die gesamte Zone, z. B. wie oft die Slave-Nameserver der Zone überprüfen sollen, ob die Zone geändert wurde. Der SOA-Record beginnt mit dem Domain-Namen der Zone, der Festlegung der Zonenklasse (fast immer *IN* für Internet) und dem Typ-Kürzel *SOA*. Hinter der Typangabe benötigt der SOA-Record sieben Felder:

#### *Das MNAME-Feld*

Geben Sie den Domain-Namen des primären Master-Nameservers der Zone an.

#### *Das RNAME-Feld*

Geben Sie eine E-Mail-Adresse an, unter der der Administrator der Zone erreicht werden kann. Ersetzen Sie das »@« in der E-Mail-Adresse durch einen Punkt (».«).

#### *Die Seriennummer der Zone*

Falls Sie die Zone nur von Hand ändern, indem Sie die Zonendaten-Datei editieren, ziehen Sie das Format *JJJJMMDDVV* in Betracht, wobei *JJJJ* für das Jahr, *MM* für den

---

<sup>2</sup> Vorausgesetzt, Sie setzen eine neuere BIND-Version als 8.2 ein – und das sollten Sie.

numerischen Monat (von 01 bis 12) und *DD* für das Tagesdatum steht; *VV* ist schließlich eine zweistellige Versionsnummer, die bei 00 beginnt. Dieses Format ist praktisch als Hinweis, wann Sie die Zone zuletzt aktualisiert haben.

#### *Der Refresh-Wert der Zone*

Dieser Wert legt fest, wie häufig die Slave-Nameserver der Zone auf ihrem Master-Nameserver überprüfen sollen, ob die Seriennummer der Zone erhöht wurde (was anzeigt, dass sich die Zone geändert hat). Dieser Wert ist nicht sonderlich wichtig, wenn Sie die NOTIFY-Methode verwenden, die es Ihrem primären Master-Nameserver ermöglicht, den Slaves *mitzuteilen*, dass die Zone geändert wurde, aber Werte zwischen einer und drei Stunden sind üblich.

#### *Der Retry-Wert der Zone*

Dieser Wert legt fest, wie oft die Slave-Nameserver der Zone ihren Master-Nameserver erneut kontaktieren sollen, nachdem die Überprüfung der Seriennummer fehlgeschlagen ist. Genau wie Refresh ist auch dieser Wert nicht so wichtig, wenn Sie NOTIFY verwenden, aber üblich sind Werte zwischen 15 Minuten und einer Stunde.

#### *Der Expire-Wert der Zone*

Dieser Wert legt fest, wie lange die Slave-Nameserver der Zone weiterhin antworten sollen, wenn sie nicht in der Lage sind, ihren Master-Nameserver zu erreichen, um die aktuelle Seriennummer zu ermitteln. Da dies festlegt, wie lange Ihre Slaves Abfragen für den Fall eines Ausfalls beantworten, sollten Sie eine recht lange Zeit einstellen. Werte von mehreren Wochen bis zu einem Monat sind üblich.

#### *Der Negative-Caching-TTL-Wert der Zone*

Dieser Wert bestimmt, wie lange andere Nameserver negative Antworten im Cache halten können, die die autorisierten Nameserver der Zone ausgegeben haben. Eine solche negative Antwort ist *NXDOMAIN*. Sie gibt an, dass der Domain-Name, den der entfernte Server abgefragt hat, in der Zone nicht existiert. Dieser Wert sollte ziemlich klein sein, zwischen 15 Minuten und 3 Stunden.

Hier sehen Sie ein Beispiel für einen SOA-Record für die Zone *foo.example*:

```
foo.example.    IN      SOA     ns1.foo.example.  (  
                hostmaster.foo.example.  
                2002040700  
                1h  
                15m  
                30d  
                1h )
```

Da in der ersten Zeile des Records nicht genügend Platz vorhanden ist, haben wir die Zeile mit »« abgeschlossen, was den Nameserver anweist, sämtlichen Text zwischen »« und der nächsten »« so zu behandeln, als sei es nur eine einzelne Zeile. (Wir könnten stattdessen auch versuchen, den gesamten Record in einer einzigen Zeile zu halten, das wäre allerdings schwierig zu lesen.)

Fügen Sie zu guter Letzt NS-Records hinzu, die die Domain-Namen der autorisierten Nameserver für die Zone auflisten. Sie haben diese Nameserver wahrscheinlich bei der Registrierung Ihres Domain-Namens festgelegt.

```
foo.example.    IN    NS    ns1.foo.example.  
foo.example.    IN    NS    ns2.foo.example.  
foo.example.    IN    NS    ns.isp.net.
```

## Erläuterung

Alle Domain-Namen in Resource Records enden mit Punkten, um den Nameserver daran zu hindern, den Ursprung anzuhängen. Der Standardursprung für eine Zonendaten-Datei ist einfach der Domain-Name der Zone, in diesem Fall *foo.example*, sodass wir den SOA-Record auch folgendermaßen schreiben könnten:

```
@    IN    SOA    ns1    (  
    hostmaster  
    2002040700  
    1h  
    15m  
    30d  
    1h )
```

(»@« ist die Kurzform für den »aktuellen Ursprung«.)

Da die ersten paar Resource Records in der Zone alle mit demselben Domain-Namen verknüpft sind (in unserem Beispiel *foo.example*), brauchen Sie den Domain-Namen nur für den ersten von ihnen festzulegen und können die restlichen mit Whitespace (Leerzeichen oder Tabs) beginnen:

```
@    IN    SOA    ns1    (  
    hostmaster  
    2002040700  
    1h  
    15m  
    30d  
    1h )  
  
    IN    NS    ns1.foo.example.  
    IN    NS    ns2.foo.example.  
    IN    NS    ns.isp.net.
```

Der Nameserver interpretiert Records, die mit Whitespace beginnen, als zu dem zuletzt angegebenen Domain-Namen gehörig.

## Siehe auch

Rezept 1.14 zur Erstellung einer *named.conf*-Datei, Rezept 1.15 zur Konfiguration eines primären Master-Nameservers und Kapitel 4 von *DNS und BIND*.

## 2.2 Einen Host hinzufügen

### Problem

Sie müssen einen Host zum DNS hinzufügen.

### Lösung

Fügen Sie zu den entsprechenden Zonen einen A- und einen PTR-Record für den Host hinzu (es handelt sich fast immer um zwei verschiedene Zonen: eine Forward-Mapping- und eine Reverse-Mapping-Zone). Um beispielsweise einen Host namens *host.foo.example* mit der IP-Adresse 10.0.0.1 zum DNS hinzuzufügen, könnten Sie den folgenden Record zur Zonendaten-Datei von *foo.example* hinzufügen:

```
host.foo.example.    IN    A    10.0.0.1
```

Und Sie würden den folgenden Record zur Zonendaten-Datei der Reverse-Mapping-Zone hinzufügen, bei der es sich um *10.in-addr.arpa*, *0.10.in-addr.arpa* oder *0.0.10.in-addr.arpa* handeln könnte, je nachdem, wie Sie die Administration Ihrer Reverse-Mapping-Domain aufteilen:

```
1.0.0.10.in-addr.arpa.  IN    PTR    host.foo.example.
```

### Erläuterung

Es steht Ihnen frei, den Vorteil des Ursprungs in der Datei zu nutzen, um die Resource Records abzukürzen. Wenn Sie zum Beispiel den A-Record zu einer Zeile in der Zonendaten-Datei hinzufügen, in der der Ursprung *foo.example* ist, können Sie Folgendes schreiben:

```
host    IN    A    10.0.0.1
```

Falls Sie den PTR-Record in eine Zeile einfügen, in der der Ursprung *0.0.10.in-addr.arpa* ist, können Sie Folgendes schreiben:

```
10     IN    PTR    host.foo.example.
```

Da die Standardklasse *IN* für Internet lautet, können Sie auch das *IN* weglassen.

Es ist wichtig, PTR-Records für Ihre Hosts hinzuzufügen. Ohne PTR-Records werden die Adressen Ihrer Hosts nicht in Domain-Namen umgewandelt, sodass die Hosts nicht in der Lage sind, auf Dienste zuzugreifen, die Reverse Mapping erfordern, und Ihre Netzwerk-Management-Software sie möglicherweise nicht automatisch identifizieren kann.

Es kann sein, dass Sie noch andere Records für den Host hinzufügen möchten. Falls der Domain-Name des Hosts auf der rechten Seite einer E-Mail-Adresse auftauchen könnte, fügen Sie einen MX-Record hinzu, der festlegt, wohin an diesen Host adressierte Mails geliefert werden sollen.

## Siehe auch

Rezept 2.4 zum Hinzufügen eines MX-Records; Rezept 2.9 zur Beschränkung der Zeit, wie lange ein Record im Cache gehalten werden kann, Rezept 2.10, in dem Sie etwas über den Umgang mit Multihomed-Hosts erfahren, sowie Kapitel 4 von *DNS und BIND*.

## 2.3 Einen Alias hinzufügen

### Problem

Sie müssen einen Alias von einem Domain-Namen auf einen anderen erzeugen.

### Lösung

Fügen Sie einen CNAME-Record zu der Zone hinzu, in die der Alias gehört. Um beispielsweise *a.foo.example* zu einem Alias für *b.bar.example* zu machen, fügen Sie den folgenden CNAME-Record zur Zonendaten-Datei von *foo.example* hinzu:

```
a.foo.example.    IN    CNAME    b.bar.example.
```

### Erläuterung

Beachten Sie, dass ein CNAME-Record den Alias äquivalent zum Ziel des Alias macht. Abfragen nach allen Arten von Records, die mit dem Alias verknüpft sind, bleiben zwar Abfragen nach demselben Record-Typ, sind allerdings mit dem Domain-Namen verknüpft, auf den der Alias verweist.

Sie sollten außerdem keine Aliase auf der rechten Seite anderer Record-Typen wie NS- oder MX-Records verwenden. Die Empfänger von NS- und MX-Records – Nameserver beziehungsweise Mail-Server – gehen davon aus, dass auf der rechten Seite kein Alias steht, und verarbeiten sie deshalb nicht korrekt. Die einzige Record-Art, die einen Alias auf der rechten Seite zulässt, ist der CNAME-Record selbst: Sie können einen Alias auf einen anderen Alias verweisen lassen, solange die Alias-Kette bei einem Domain-Namen endet, der kein Alias ist. Stellen Sie jedoch sicher, dass die Kette nicht länger ist als acht Verknüpfungen, und hüten Sie sich vor Alias-Schleifen.

Beachten Sie zu guter Letzt, dass der CNAME-Record in die Zone gehört, die den Domain-Namen des Alias enthält und nicht dessen Ziel.

## Siehe auch

Rezept 2.6 zur Einrichtung virtueller Web-Hosts sowie Kapitel 4 und den Abschnitt »CNAME-Records verwenden« in Kapitel 16 von *DNS und BIND*.

## 2.4 Ein Mail-Ziel hinzufügen

### Problem

Sie müssen ein Mail-Ziel zum DNS hinzufügen.

### Lösung

Fügen Sie einen oder mehrere MX-Records zu der Zone hinzu, die den Domain-Namen des Mail-Ziels enthält. Der MX-Record legt den oder die Mail-Server fest, die an dieses Ziel adressierte Mails entgegennehmen. Jeder MX-Record erfordert einen Präferenzwert, der den Mailern, die Mail versenden, die Reihenfolge mitteilt, in der die Ziel-Mail-Server kontaktiert werden sollen. Je *niedriger* der Präferenzwert ist, umso *mehr* wird der Mail-Server bevorzugt.

Um Mailer beispielsweise anzuweisen, an *foo.example* gerichtete Mails (etwa eine E-Mail-Nachricht, die an *hostmaster@foo.example* gerichtet ist) an *mail.foo.example* zu senden, und an *smtp.isp.net* nur, wenn *mail.foo.example* offline ist oder keine Verbindungen akzeptiert, fügen Sie die folgenden MX-Records zur Zonendaten-Datei von *foo.example* hinzu:

```
foo.example.    IN    MX    0 mail.foo.example.  
foo.example.    IN    MX    10 smtp.isp.net.
```

### Erläuterung

Der Präferenzwert ist eine vorzeichenlose 16-Bit-Zahl, liegt also zwischen 0 und 65.535. Die genaue Zahl ist unwichtig: Der Präferenzwert stellt keine wie auch immer geartete Maßeinheit dar. Wichtig ist lediglich, dass die Präferenzwerte für die MX-Records eines Domain-Namens einem sendenden Mailer die Reihenfolge mitteilen, in der er die Ziel-Mail-Server verwenden soll.

Die meisten Mailer werden die Last zufällig auf die Mail-Server mit demselben Präferenzwert verteilen. Dies kann bei häufig verwendeten Mail-Zielen praktisch sein: Sie können eine Reihe von Mail-Servern mit demselben Präferenzwert angeben, sodass die sendenden Mailer die Lieferung Ihrer Mail auf diese Mail-Server aufteilen.

Der Mail-Server muss als einzelner Domain-Name angegeben werden, nicht als IP-Adresse. Wenn Sie eine IP-Adresse auf der rechten Seite eines MX-Records verwenden, versuchen Mailer – die an dieser Stelle einen Domain-Namen erwarten –, die IP-Adresse als Domain-Namen nachzuschlagen. Dies führt zu unnötigen Abfragen an die Root-Nameserver und scheitert trotzdem bei dem Versuch, eine IP-Adresse zu ermitteln.

Es ist Ihre Aufgabe (oder die Ihrer Postmaster-Kollegen), die Mail-Server so zu konfigurieren, dass sie an das Ziel adressierte Mails akzeptieren. Stellen Sie sicher, dass die wich-

tigste Mail-Exchanger verstehen, dass das Mail-Ziel lokal ist, und dass seltener verwendete Mail-Exchanger so konfiguriert werden, dass sie an das Ziel gerichtete Mails weiterleiten.

## Siehe auch

RFC 2821, das autorisierte Informationen über SMTP und die Verwendung von MX-Records enthält, sowie Kapitel 5 von *DNS und BIND*.

## 2.5 Den Domain-Namen Ihrer Zone auf Ihren Webserver verweisen lassen

### Problem

Sie möchten, dass der Domain-Name Ihrer Zone auf Ihren Webserver verweist.

### Lösung

Fügen Sie einen A-Record zu dem Domain-Namen Ihrer Zone hinzu, der auf die IP-Adresse Ihres Webserver verweist:

```
foo.example.    IN    A    10.0.0.1
```

### Erläuterung

Das Eintragen eines solchen A-Records sorgt dafür, dass die Benutzer einfach *http://foo.example/* eingeben (und das führende »www« weglassen) können, wenn sie auf Ihre Website zugreifen. Einige populäre Websites veröffentlichen ihre URLs in dieser Form, etwa CNN.

Viele Administratoren versuchen, dieses Problem zu lösen, indem sie statt des A-Records einen CNAME-Record für den Domain-Namen der Zone hinzufügen:

```
foo.example.    IN    CNAME    www.foo.example.
```

Dies ist jedoch nicht erlaubt, weil es der Regel widerspricht, dass mit einem Alias keine Records außer einem CNAME-Record verknüpft sein dürfen.

Falls Sie mehrere Webserver betreiben, können Sie mehrere A-Records für den Domain-Namen Ihrer Zone hinzufügen:

```
foo.example.    IN    A    10.0.0.1
foo.example.    IN    A    10.0.0.2
foo.example.    IN    A    10.0.0.3
```

Diese Records werden standardmäßig in Round-Robin-Reihenfolge ausgegeben, die in Rezept 2.7 beschrieben wird.

## Siehe auch

Rezept 2.3, das weitere Informationen über CNAME-Records enthält, Rezept 2.6, das beschreibt, wie Sie von einem Domain-Namen auf eine bestimmte URL und nicht nur auf einen bestimmten Webserver verweisen, und Rezept 2.7, das das Round-Robin-Verfahren beschreibt.

## 2.6 Von einem Domain-Namen auf eine bestimmte URL verweisen

### Problem

Sie möchten, dass die Benutzer, die auf einen Ihrer Domain-Namen zugreifen, eine bestimmte URL erreichen.

### Lösung

Fügen Sie einen A-Record zu der Zone hinzu, zu der der Domain-Name gehört, auf den die IP-Adresse des Webserver verweist:

```
mylink.foo.example.    IN    A    10.0.0.1
```

Konfigurieren Sie anschließend den Webserver so, dass er Browser, die *http://mylink.foo.example* anfordern, auf das entsprechende Verzeichnis auf Ihrem Webserver umleitet.

### Erläuterung

Bei dieser Lösung findet der Großteil der Konfiguration auf dem Webserver statt; das Hilfsmittel wird »virtuelle Hosts« genannt. Der Webserver muss Ihren Domain-Namen, der im HTTP/1.1-»Host«-Header erscheint, mit einer bestimmten »Document Root« verknüpfen, einem Verzeichnis im Dokumentenbaum des Webserver.

Falls der Domain-Name auf dem Webserver eine Zone ist, die von jemand anderem betrieben wird, oder falls Sie bereits einen Domain-Namen in Ihrer Zone auf die Adresse des Webserver zeigen lassen, können Sie einen CNAME-Record statt eines A-Records verwenden:

```
mylink.foo.example.    IN    CNAME    www.isp.net.
```

Auf diese Weise verweist Ihre Domain bei einer Änderung der IP-Adresse des Webserver weiterhin auf die richtige Position.

Falls der Webserver von jemand anderem betrieben wird, benötigen Sie natürlich dessen Mitarbeit, um die Verknüpfung zwischen *mylink.foo.example* und dem entsprechenden Verzeichnis einzurichten.

## Siehe auch

Rezept 2.5 erklärt, wie man einen Domain-Namen auf einen Webserver verweisen lässt. Siehe auch die Online-Dokumente der Apache Software Foundation über virtuelle Hosts unter <http://httpd.apache.org/docs/vhosts/name-based.html> und <http://httpd.apache.org/docs-2.0/vhosts/> sowie Kapitel 4 von *Apache: Das umfassende Handbuch*, »Virtuelle Hosts«.

## 2.7 Die Round-Robin-Lastverteilung

### Problem

Sie möchten Round-Robin für einen Domain-Namen einrichten.

### Lösung

Fügen Sie einfach mehrere A-Records für den Domain-Namen hinzu. Zum Beispiel:

```
www.foo.example.    IN    A    10.0.0.1
www.foo.example.    IN    A    10.0.0.2
www.foo.example.    IN    A    10.0.0.3
```

In aufeinander folgenden Antworten auf Abfragen der Adresse von *www.foo.example* vertauschen die *foo.example*-Nameserver die Reihenfolge, in der sie die A-Records zurückgeben, indem sie nach jeder Antwort den ersten A-Record an das Ende der Liste setzen.

### Erläuterung

Alle modernen Nameserver geben Resource Records standardmäßig in Round-Robin-Reihenfolge aus. Nur sehr alte Nameserver (vor BIND 4.9) unterstützen Round-Robin nicht.

Denken Sie daran, dass Round-Robin kein *Load-Balancing* ist. Der Nameserver hat keine Ahnung, wie viel die Webserver, die den Content von *www.foo.example* liefern, zu tun haben, und wissen noch nicht einmal, ob sie alle antworten. Sollte der Webserver unter 10.0.0.1 abstürzen, würde der Nameserver dennoch jedes dritte Mal seine Adresse ausgeben. Für echtes Load-Balancing brauchen Sie etwas mehr als nur DNS.

## Siehe auch

Rezept 3.18 enthält Details über die Arbeitsweise von Round-Robin und zeigt, wie Sie es abschalten können.

## 2.8 Einen Domain-Namen zu einer Subdomain hinzufügen, ohne eine neue Zone zu erzeugen

### Problem

Sie möchten einen Domain-Namen zu einer Subdomain Ihrer Zone hinzufügen, aber keine neue Zone einrichten und von Ihrer aktuellen Zone delegieren.

### Lösung

Fügen Sie einfach die mit dem neuen Domain-Namen verknüpften Records hinzu, und legen Sie die Subdomain im Domain-Namen fest. Um beispielsweise den Domain-Namen *a.b.foo.example* zu der Zone *foo.example* hinzuzufügen, könnten Sie den folgenden Record in die Zonendaten-Datei von *foo.example* einfügen:

```
a.b.foo.example.    IN    A    10.0.0.4
```

Die implizite Erwähnung erzeugt die Subdomain *b.foo.example* und den Domain-Namen *a.b.foo.example*. Die Subdomain *b.foo.example* ist ein Teil der Zone *foo.example* (genau wie der Domain-Name *a.b.foo.example*) und wird in Transfers der Zone an die Slave-Nameserver einbezogen.

Falls der Standardursprung in der Zonendaten-Datei *foo.example* ist, können Sie den Record auch so schreiben:

```
a.b                IN    A    10.0.0.4
```

### Erläuterung

Manchmal ist die Lösung für ein Problem einfach die offensichtlichste Möglichkeit von allen. Das ist sowohl beim Einrichten von Round-Robin als auch bei diesem Problem der Fall. Aber viele Administratoren – selbst die sehr erfahrenen – sind nicht damit vertraut, Domain-Namen zu ihren Zonen hinzuzufügen, die links von den Domain-Namen ihrer Zone mehrere Labels haben. Sie stellen sich die Domain-Namen in ihren Zonen immer in dem Format *host.domain-name-of-zone* vor und nicht als beliebige Anzahl von Labels, die mit dem Domain-Namen der Zone enden.

### Siehe auch

Für Näheres über Intra-Zonen-Subdomains siehe den Abschnitt »Anlegen einer Subdomain in der Zone des Parents« in Kapitel 9 von *DNS und BIND*. Wenn Sie die Subdomain delegieren und eine neue Zone einrichten möchten, siehe Rezept 6.1.

## 2.9 Entfernte Nameserver vom Caching eines Resource Records abhalten

### Problem

Sie möchten entfernte Nameserver davon abhalten, einen oder mehrere Records in Ihrer Zone im Cache zu halten.

### Lösung

Weisen Sie dem Record (oder den Records) explizit eine – kurze – Time-to-live (TTL) zu. Um beispielsweise andere Server daran zu hindern, die Adresse Ihres Webservers im Cache zu halten, könnten Sie die folgenden A-Records zur Zonendaten-Datei hinzufügen:

```
www.foo.example. 1 IN A 10.0.0.1
www.foo.example. 1 IN A 10.0.0.2
www.foo.example. 1 IN A 10.0.0.3
```

Geben Sie die explizite TTL zwischen dem Domain-Name-Eigentümer des Records und dem Klassen-Feld an. Standardmäßig ist der Wert eine ganzzahlige Sekundenzahl. Sie können auch einen Wert mit Maßeinheit angeben, genau wie bei der *\$TTL*-Steueranweisung.

### Erläuterung

Beachten Sie, dass die TTLs für die drei *www.foo.example*-A-Records gleich sind; das ist kein Versehen. Sollten Sie verschiedene TTLs für Records im selben RR-Satz (vom gleichen Typ und mit demselben Domain-Namen verknüpft) verwenden, könnte es sein, dass ein entfernter Nameserver nur manche von ihnen ungültig werden lässt, was zu unvorhersagbaren Ergebnissen führen kann. Folglich beachten moderne Nameserver diese Fehlkonfiguration und »stutzen« nicht übereinstimmende TTLs innerhalb desselben RR-Satzes auf die kleinste TTL der Gruppe.

Warum habe ich die TTL eins und nicht null verwendet? Immerhin scheint eine Null-TTL auszusagen: »Halte diesen Record nicht im Cache.« Unglücklicherweise rufen TTLs von null bei einigen älteren Nameservern einen Bug hervor, wodurch die Records ungültig werden, bevor sie an den Resolver zurückgegeben werden, der die Abfrage ausgesendet hatte. Oh je!

### Siehe auch

Rezept 2.1, das die Syntax von Werten mit Maßeinheit beschreibt und »TTLs ändern« in Kapitel 8 von *DNS und BIND*.

## 2.10 Einen Multihomed-Host hinzufügen

### Problem

Sie möchten einen Multihomed-Host zum DNS hinzufügen.

### Lösung

Fügen Sie mehrere A-Records für den Domain-Namen des Hosts hinzu – für jede IP-Adresse einen. Für einen Fileserver mit zwei Netzwerkschnittstellen könnten Sie beispielsweise die folgenden Records hinzufügen:

```
fs.foo.example.    IN    A    10.0.0.9
fs.foo.example.    IN    A    192.168.0.9
```

Um das Reverse Mapping für den Host zu regeln, fügen Sie zu jeder der beiden entsprechenden Reverse-Mapping-Zonen einen PTR-Record hinzu:

```
9.0.0.10.in-addr.arpa.  IN    PTR    fs.foo.example.
```

und

```
9.0.168.192.in-addr.arpa.  IN    PTR    fs.foo.example.
```

### Erläuterung

Clients, die die Adresse von *fs.foo.example* nachschlagen, sehen beide IP-Adressen und können auswählen, welche sie verwenden möchten (obwohl die meisten Clients nur die erste zurückgegebene Adresse verwenden werden). Beachten Sie, dass sie standardmäßig in Round-Robin-Reihenfolge zurückgegeben werden.

Für Troubleshooting-Zwecke sollten Sie möglicherweise zwei weitere A-Records hinzufügen, von denen jeder auf eine der Adressen Ihres Multihomed-Hosts umgesetzt wird. Zum Beispiel:

```
fs-eth0.foo.example.  IN    A    10.0.0.9
fs-eth1.foo.example.  IN    A    192.168.0.9
```

So können Sie überprüfen, ob eine bestimmte Netzwerkschnittstelle auf dem Fileserver bereit ist, indem Sie beispielsweise *fs-eth0.foo.example* mit Ping ansprechen. Sie sollten jedoch in der Regel keine PTR-Records hinzufügen, die die Adressen wieder mit diesen schnittstellenspezifischen Namen zurückverbinden: Die meisten Programme können nicht mit mehreren Reverse Mappings für eine einzelne IP-Adresse umgehen.

### Siehe auch

Rezept 2.7 zum Verhalten von Round-Robin und Kapitel 4 von *DNS und BIND*.

## 2.11 Die Root-Hints-Datei eines Nameservers aktualisieren

### Problem

Sie müssen die Root-Hints-Datei eines Nameservers aktualisieren.

### Lösung

Laden Sie sich per FTP eine Kopie der neuesten Root-Hints-Datei von *ftp.rs.internic.net* herunter. Sie heißt *named.root* und liegt im Verzeichnis *domain*.

### Erläuterung

Die Root-Hints-Datei, die einem Nameserver die Domain-Namen und Adressen der Root-Nameserver mitteilt, braucht nicht oft aktualisiert zu werden. Die »aktuelle« Version stammt aus dem August 1997, und die Datei kann ohne nachteilige Wirkung leicht veraltet sein.

Wenn Sie eine neue Root-Hints-Datei herunterladen, sollten Sie daran denken, den Namen der Datei so zu ändern, wie Sie ihn in Ihrer *zone*-Anweisung für die Root-Hints angegeben haben. Starten Sie dann den Nameserver neu.

### Siehe auch

»Die Root-Hints-Daten« in Kapitel 4 und »Die Root-Hints pflegen« in Kapitel 7 von *DNS und BIND*.

## 2.12 Eine einzelne Datendatei für mehrere Zonen verwenden

### Problem

Sie möchten eine einzelne Datendatei für mehrere Zonen verwenden.

### Lösung

Erzeugen Sie eine »template«-Zonendaten-Datei. Achten Sie darauf, dass alle Eigentümernamen der Records in der Zone »@« lauten (Kurzform für den Ursprung) oder relativ sind, das heißt ohne Punkt am Ende. Zum Beispiel:

```
@ IN SOA ns1.isp.net. hostmaster.isp.net. (
2002040900
```

```

3600
900
604800
3600 )

IN NS ns1.isp.net.
IN NS ns2.isp.net.

IN MX smtp.isp.net.

IN A 192.168.0.99

www IN CNAME @

```

Fügen Sie *zone*-Anweisungen zur *named.conf*-Datei Ihres Nameservers hinzu, konfigurieren Sie sie als primären Master für die verschiedenen Zonen, und geben Sie in der *file*-Unteranweisung jedes Mal die »template«-Zonendaten-Datei an. Zum Beispiel:

```

zone "foo.example" {
    type master;
    file "db.template";
};

zone "bar.example" {
    type master;
    file "db.template";
};

zone "baz.example" {
    type master;
    file "db.template";
};

```

Da jede *zone*-Anweisung den Standardursprung in der Datendatei auf den Domain-Namen der Zone setzt, werden der SOA-Record und die NS-Records stets mit dem richtigen Domain-Namen verknüpft, und die restlichen Records werden auf die Zone »umgesetzt«.

## Erläuterung

Dieses Verfahren funktioniert nur, wenn alle Zonen einander sehr ähnlich sind – eigentlich sogar fast identisch. Die Zonen müssen die gleiche Anzahl und Zusammensetzung von Records enthalten, und die Records in den Zonen dürfen sich nur durch den Domain-Namen der Zone unterscheiden. Wenn die Domain *www.foo.example* zum Beispiel ein Alias von *a.foo.example* in der Zone *foo.example* ist, dann wird *www.bar.example* auch ein Alias von *a.bar.example* in der Zone *bar.example*.

Der Nameserver muss der primäre Master für alle diese Zonen sein. Es besteht keine Möglichkeit, einen äquivalenten Slave-Nameserver einzurichten, der dieselbe Backup-Zonendaten-Datei für alle seine Zonen verwendet, da Nameserver vollqualifizierte Domain-Namen in Backup-Zonendaten-Dateien schreiben.

Außerdem kann keine der Zonen dynamisch aktualisiert werden, da dynamische Updates einer Zone dazu führen würden, dass der Nameserver die Zonendaten-Datei neu schreibt, und die neu geschriebene Zonendaten-Datei würde ebenfalls vollqualifizierte Domain-Namen enthalten.

## Siehe auch

Rezept 2.1, in dem der Standardursprung einer Zonendaten-Datei erläutert wird.

## 2.13 Mehrere Datendateien für eine einzelne Zone

### Problem

Sie möchten eine Zone in mehrere Datendateien unterteilen, z. B., um die große Anzahl der Resource Records logisch zu organisieren.

### Lösung

Verwenden Sie die Steueranweisung *\$INCLUDE* in der Zonendaten-Datei Ihrer Parent-Zone, die den Inhalt einer anderen Datei interpoliert. Um zum Beispiel den Inhalt der Datei *db.foo.example.hosts* in die Datendatei für die Zone *foo.example* einzubinden, könnten Sie die folgende *\$INCLUDE*-Steueranweisung verwenden:

```
$INCLUDE db.foo.example.hosts
```

### Erläuterung

Der Ursprung in der eingefügten Datei ist standardmäßig derselbe wie der Ursprung in der Datei, in die sie eingefügt wird. Falls Sie den Ursprung in der eingefügten Datei ändern möchten, geben Sie den neuen Ursprung als zweites Argument der *\$INCLUDE*-Steueranweisung an:

```
$INCLUDE db.subdomain.foo.example.hosts subdomain.foo.example
```

In der Zeile hinter der *\$INCLUDE*-Anweisung wird der Ursprung automatisch auf seine vorherige Einstellung zurückgesetzt.

## Siehe auch

Rezept 2.8, das erläutert, wie eine Subdomain innerhalb derselben Zone erzeugt wird.

## 2.14 Die Seriennummer Ihrer Zone zurücksetzen

### Problem

Sie müssen Ihre Seriennummer auf einen niedrigeren Wert zurücksetzen, z. B., weil Sie unabsichtlich eine Ziffer hinzugefügt haben.

### Lösung

Falls Sie Ihre Seriennummer versehentlich auf einen höheren Wert als  $2^{32} - 1$  (4.294.967.295) erhöht haben, ermitteln Sie als Erstes Ihre aktuelle Seriennummer – wahrscheinlich ist es nicht diejenige, die Sie vermuten (die Seriennummer ist nur 32 Bit groß). Die einfachste Möglichkeit dafür besteht darin, ein Abfrage-Tool wie *dig* zu verwenden, um den SOA-Record Ihrer Zone nachzuschlagen:

```
$ dig soa foo.example

; <<>> DiG 9.2.1 <<>> soa foo.example
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4335
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;foo.example.                IN      SOA

;; ANSWER SECTION:
foo.example.                86400  IN      SOA      ns1.foo.example. hostmaster.foo.example.
2002021239 3600 900 2592000 3600
```

Falls die aktuelle Seriennummer kleiner als 2.147.483.647 ist, addieren Sie 2.147.483.647 dazu. Warten Sie, bis alle Slave-Nameserver Ihrer Zone die neue Version der Zone übernommen haben (falls Sie NOTIFY verwenden, sollte das nicht lange dauern). Setzen Sie die Seriennummer anschließend auf das gewünschte Ziel.

Wenn die aktuelle Seriennummer größer als 2.147.483.647 ist, setzen Sie sie einfach auf die gewünschte Zahl.

### Erläuterung

Wie bitte? Warum um alles in der Welt funktioniert das?

Nameserver vergleichen Seriennummern mit Hilfe der *Raumfolgenarithmetik*, die sich von Großvaters Arithmetik unterscheidet. In der Raumfolgenarithmetik haben Sie eine endliche Menge ganzer Zahlen, aber jede Zahl besitzt einen »Nachfolger«. Nach 0 kommt 1, dann 2, bis hin zu 4.294.967.295 ( $2^{32} - 1$ ). Die nächste Zahl nach 4.294.967.295 ist 0. Stellen Sie sich das wie bei einer Uhr vor: Die Stunde nach 1:00 ist 2:00 und die Stunde nach 12:00 ist 1:00.

Die Hälfte der Zahlen sind größer als jede angegebene Zahl, und die andere Hälfte ist kleiner. Bei einer Anzahl von  $2^{32}$  möglichen Seriennummern ist die Hälfte (oder genauer gesagt  $2^{31} - 1$ ) größer als jede beliebige Seriennummer und die Hälfte kleiner.

Betrachten Sie die Seriennummer 1.000.000.000. Die nächsten  $2^{31} - 1$  Seriennummern, 1.000.000.001 bis 3.147.483.647, sind größer. Die darauf folgenden  $2^{31} - 1$  Seriennummern, 3.147.483.648 bis 4.294.967.295 ( $2^{32} - 1$ ) und 0 bis 999.999.999, sind kleiner. Ja, Alice, im Wunderland der Seriennummern ist 3.147.483.648 *kleiner* als 1.000.000.000.

Wenn Sie also 2.147.483.647 ( $2^{31} - 1$ ) zu einer Seriennummer addieren, dann addieren Sie das größte mögliche Inkrement hinzu. Falls Sie eine noch größere Zahl addieren, wird das Ergebnis in Wirklichkeit *kleiner* als die alte Seriennummer, und die Slaves Ihrer Zone laden die Zone nicht.

Nachdem alle Slaves die neue Zone erhalten haben, können Sie die Seriennummer einfach auf den gewünschten Wert setzen, der nun als größer betrachtet wird als die aktuelle Seriennummer.

Wenn Sie mit dieser Neuen Mathematik nicht vertraut sind, probieren Sie das Skript `reset_serial.pl` aus, das sich in der `tar`-Datei zu diesem Buch befindet (im Vorwort finden Sie die Adresse, wo Sie sie erhalten). Ihre Slave-Nameserver haben keine andere Wahl, als die Zone zu laden, ungeachtet ihrer Seriennummer.

Dies funktioniert natürlich nicht, wenn Sie keine administrative Kontrolle über all Ihre Slaves ausüben, und das ist in etwa so elegant, als würden Sie einen Schlitzschraubendreher als Meißel verwenden.

## Siehe auch

»Neustart mit einer neuen Seriennummer« in Kapitel 7 von *DNS und BIND* sowie RFC 1982, das die Seriennummern-Arithmetik erläutert.

## 2.15 Manuelle Änderungen an einer dynamisch aktualisierten Zone vornehmen

### Problem

Sie möchten eine Zonendaten-Datei von Hand editieren, aber die Zone wird dynamisch aktualisiert.

### Lösung

Wenn Sie einen BIND 8-Nameserver verwenden, halten Sie den Nameserver mit `ndc stop` an. Löschen Sie die Logdatei der dynamischen Zonen-Updates (deren Name standardmäßig genauso lautet wie der Name der Zonendaten-Datei mit angehängtem `.log`) und

die IXFR-Logdatei, falls vorhanden (ihr Name entspricht der Zonendaten-Datei plus *.ixfr*). Editieren Sie dann die Zonendaten-Datei, und starten Sie den Nameserver wieder.

Wenn Sie BIND 9 einsetzen, halten Sie den Nameserver mit *rndc stop* an, löschen die Journaldatei der Zone (deren Name dem Namen der Zonendaten-Datei mit *.jnl* am Ende entspricht), editieren die Zonendaten-Datei und starten den Nameserver wieder.

Auf einem BIND 9.3.0-Nameserver oder neuer können Sie die Zone mit *rndc freeze* einfrieren, die Zonendaten-Datei editieren und das Einfrieren der Zone mit *rndc unfreeze* wieder aufheben.

## Erläuterung

Bei dynamischen Zonen ist es besser, alle Änderungen an der Zone mit Hilfe dynamischer Updates vorzunehmen. Allerdings ist das oft nicht praktisch.

Das Problem besteht bei den meisten BIND-Nameservern darin, dass Ihre Änderungen verloren gehen können, wenn Sie eine Zonendaten-Datei editieren, während der Server läuft. Wenn Sie den Nameserver neu starten (das erneute Laden dynamischer Zonen funktioniert nicht), schreibt der Nameserver die Zonendaten-Datei neu, falls er irgendwelche dynamischen Updates der Zone empfangen hat, die noch nicht in die Zonendaten-Datei geschrieben wurden. Was passiert mit Ihren Änderungen? Die verschwinden spurlos, genau wie so viele Dotcoms. Sie müssen den Nameserver anhalten, bevor Sie die Zonendaten-Datei editieren. Und das bedeutet, dass Ihr Nameserver dynamische Updates verpassen könnte, während Sie die Zonendaten-Datei von Hand editieren, also beeilen Sie sich!

Falls Sie die Zonendaten-Datei manuell editieren, werden die Änderungen, die Sie durchführen, außerdem nicht in die Logdatei der dynamischen Updates eingetragen werden – in die *.log*-Datei bei BIND 8 und in die *.jnl*-Datei bei BIND 9. Wenn der Nameserver die Zonendaten-Datei lädt und anschließend den Inhalt der Logdatei überprüft, entdeckt er eine Lücke: Ihm fehlt der Eintrag für die letzte Änderung, diejenige, die Sie manuell durchgeführt haben. Deshalb müssen Sie vor dem Laden die Logdatei löschen.

Der Preis für das Löschen der Logdatei ist, dass die Slaves Ihrer Zone bei ihrem nächsten Versuch keinen inkrementellen Zonentransfer erhalten werden, da der Eintrag der letzten Änderung – er ist für das Update notwendig – fehlt. Sie werden einen inkrementellen Zonentransfer anfordern, aber stattdessen einen vollständigen Zonentransfer erhalten.

Der BIND 9.3.0-Nameserver bietet die beiden neuen *rndc*-Befehle *freeze* und *unfreeze*, die es Ihnen ermöglichen, die Verarbeitung der dynamischen Updates einer Zone anzuhalten und wieder aufzunehmen. *freeze* löscht außerdem die Logdatei. Entsprechend können Sie die Zone mit *rndc freeze* einfrieren, die Zonendaten-Datei editieren und das Einfrieren mit *rndc unfreeze* wieder aufheben.

## Siehe auch

Rezept 5.19, in dem Sie die Verwendung des Programms *nsupdate* zum Modifizieren einer Zone kennen lernen.

## 2.16 Die Adresse eines Hosts ändern

### Problem

Sie möchten einen Host von einer Adresse auf eine andere verschieben.

### Lösung

Spätestens eine TTL-Phase vor der Änderung sollten Sie die TTL für den A-Record und den PTR-Record des Hosts auf einen niedrigen Wert reduzieren, etwa auf 60 Sekunden. Nehmen wir zum Beispiel an, Sie planen, den Host *z.foo.example* zu verschieben. Falls sein aktueller A-Record so aussieht:

```
z.foo.example.      86400   IN      A       192.168.0.254
```

reduzieren Sie die TTL mindestens einen Tag (86.400 Sekunden) vor der Aktion, und zwar folgendermaßen:

```
z.foo.example.      60      IN      A       192.168.0.254
```

Reduzieren Sie gleichzeitig die TTL des PTR-Records für den Host:

```
254.0.168.192.in-addr.arpa. 60      IN      PTR     z.foo.example.
```

Nachdem Sie den Host dann verschoben haben, ändern Sie den A-Record so, dass er die neue Adresse des Hosts wiedergibt, und stellen die TTL wieder her:

```
z.foo.example.      86400   IN      A       10.0.0.254
```

Löschen Sie den alten PTR-Record, und fügen Sie (zur passenden Zonendaten-Datei!) für die neue Adresse wieder einen hinzu:

```
254.0.0.10.in-addr.arpa. 86400   IN      PTR     z.foo.example.
```

### Erläuterung

Sie müssen die TTL der alten Records vorzeitig reduzieren, um Nameserver daran zu hindern, diese noch vor der Aktion in den Cache aufzunehmen. Wenn Sie die TTL einfach unverändert ließen, könnte ein entfernter Nameserver die alte Adresse kurz vor der Änderung in den Cache aufnehmen, und es würde einige Zeit dauern, bis dieser Record ungültig würde. Falls Sie nicht NOTIFY verwenden, sollten Sie auch die Refresh-Zeit der Zonen einrechnen, in denen die Records sich befinden, da es so lange dauern könnte, bis die Records mit der niedrigeren TTL alle Ihre Slaves erreicht haben.

Natürlich lässt sich dieses Verfahren nicht nur auf A- und PTR-Records anwenden. Sie könnten es genauso einfach zur Änderung von MX-Records oder für jeden anderen Record-Typ verwenden. Falls es sich bei dem verschobenen Host allerdings um einen Nameserver handelt oder falls Sie die NS-Records Ihrer Zone ändern müssen, sollten Sie sich die Rezepte 6.6 und 6.7 ansehen.

Beachten Sie, dass der neue PTR-Record häufig in eine andere Zone gehören kann als der alte.

## Siehe auch

Rezept 2.9 mit einer Erläuterung, wie die TTL eines einzelnen Records reduziert wird; die Rezepte 6.6 und 6.7 zum Verschieben eines Nameservers und zur Änderung aller Nameserver einer Zone sowie »TTLs ändern« in Kapitel 8 von *DNS und BIND*.

## 2.17 Alle Domain-Namen in einer Zone auf eine einzige IP-Adresse umsetzen

### Problem

Sie möchten jeden Domain-Namen in einer Zone auf eine einzige IP-Adresse umsetzen.

### Lösung

Fügen Sie zu der Zone einen A-Record hinzu, der mit dem Wildcard-Domain-Namen verknüpft ist. Zum Beispiel:

```
*.foo.example.    IN    A    10.0.0.1
```

### Erläuterung

Technisch gesehen, setzt dieser Record nicht *jeden* Domain-Namen in der Zone auf 10.0.0.1 um. In Wirklichkeit bezieht sich der Wildcard-Domain-Name nicht auf Domain-Namen in der Zonendaten-Datei. Angenommen, Sie hätten auch den Domain-Namen *ns1.foo.example* in der Zone *foo.example*:

```
ns1.foo.example.  IN    A    192.168.0.1
```

dann würden Abfragen nach der Adresse von *ns1.foo.example* dem Wildcard-Domain-Namen *nicht* entsprechen, was auch wünschenswert ist, da *ns1.foo.example* eine andere Adresse hat. Der Wildcard-Domain-Name würde sich auch nicht auf Domain-Namen beziehen, die Eigentümer anderer Arten von Records sind. Zum Beispiel könnten Sie den folgenden Record in der Zone haben:

```
text.foo.example. IN    TXT    "Textkommentar"
```

Abfragen der Adresse von *text.foo.example* würden eine leere Antwort zurückgeben, da *text.foo.example* keine Adresse hat.

Worauf *bezieht* sich denn nun der Wildcard-Domain-Name? Sie beziehen sich auf Abfragen von Domain-Namen in der Zone, die nicht in der Zonendaten-Datei auftauchen, das heißt, auf jeden vorstellbaren Domain-Namen, der mit *foo.example* endet, nicht in der *foo.example*-Zonendaten-Datei steht und auch kein Teil einer delegierten Subdomain von *foo.example* ist.

Wildcard-Domain-Namen können auch Eigentümer anderer Typen von Records sein. Betrachten Sie zum Beispiel diesen CNAME-Record:

```
*.foo.example.    IN    CNAME    foo.example.
```

Er erzeugt einen Alias für jeden Domain-Namen in der Zone, für den keine expliziten Records mit dem Domain-Namen *foo.example* verknüpft sind. Falls Sie also explizite Records für *www.foo.example* weglassen, würde jemand, der *www.foo.example* nachschlägt, erfahren, dass dieser Domain-Name ein Alias von *foo.example* ist. Wer *zaphod.beeblebrox.foo.example* nachschlägt, würde erfahren, dass auch das ein Alias von *foo.example* ist – natürlich nur unter der Voraussetzung, dass Sie mit dem Domain-Namen *zaphod.beeblebrox.foo.example* keine Records verknüpft haben. Deshalb können Sie sich ein Wildcard als »Standard«-Domain-Namen für eine Zone vorstellen: Jeder explizite Domain-Name in der Zone besitzt nur die Records, die Sie für ihn anlegen, aber das Wildcard bezieht sich auf jede andere Domain in der Zone.

Wie das Beispiel *zaphod.beeblebrox.foo.example* bereits vermuten lässt, können Wildcards mehr als einem Label entsprechen. Genauer gesagt kann ein Wildcard null oder mehr Labels entsprechen. Allerdings würde der Wildcard-Domain-Name in dem CNAME-Record nicht *foo.example* entsprechen, da *\*.foo.example* sogar mit null Labels einen Punkt mehr besitzt als *foo.example*.

## Siehe auch

»Wildcards« in Kapitel 16 von *DNS und BIND*.

## 2.18 Ähnliche Records hinzufügen

### Problem

Sie möchten eine Reihe von Records hinzufügen, die sich nur leicht voneinander unterscheiden.

### Lösung

Verwenden Sie die Steueranweisung *\$GENERATE*, um eine Vorlage anzugeben, die der Nameserver zur Erzeugung einer Reihe ähnlicher Records verwenden soll. Um zum Bei-

spiel eine Reihe von PTR-Records hinzuzufügen, die sich nur durch eine einzelne Ziffer unterscheiden, könnten Sie die folgende *\$GENERATE*-Steueranweisung verwenden:

```
$GENERATE 11-20 $.0.168.192.in-addr.arpa. PTR dhcp-$.foo.example.
```

Ihr BIND-Nameserver liest den Bereich (11 – 20) und die Vorlage (*\$.0.168.192.in-addr.arpa. PTR dhcp-\$.foo.example.*) aus der *\$GENERATE*-Steueranweisung. Anschließend geht er den Bereich schrittweise durch und ersetzt jedes Dollar-Zeichen (»\$«) in der Vorlage durch den aktuellen Wert, was zehn PTR-Records erzeugt:

```
11.0.168.192.in-addr.arpa. PTR dhcp-11.foo.example.  
12.0.168.192.in-addr.arpa. PTR dhcp-12.foo.example.  
13.0.168.192.in-addr.arpa. PTR dhcp-13.foo.example.  
...  
20.0.168.192.in-addr.arpa. PTR dhcp-20.foo.example.
```

## Erläuterung

*\$GENERATE* unterstützt nur eine beschränkte Menge von Record-Typen: A, AAAA, CNAME, DNAME, NS und PTR. Außerdem kann die Vorlage kein TTL- oder Klassenfeld enthalten, nur einen Typ.

Mit etwas mehr Phantasie können Sie den Bereich auch in dem Format *Start-Ende/Schrittweite* durchgehen. So würde *0–100/2* etwa in Zweierschritten von 0 bis 100 zählen.

BIND 8.2 hat *\$GENERATE* erstmals der Welt vorgestellt. BIND 9.1.0 hat *\$GENERATE* in die BIND 9-Releases eingeführt.

Beachten Sie, dass *\$GENERATE* anders als die Steueranweisungen *\$INCLUDE* und *\$ORIGIN* nur von BIND-Nameservern unterstützt wird; Sie können es zum Beispiel nicht in einer Zonendaten-Datei auf einem Microsoft-DNS-Server verwenden.

## Siehe auch

»Aufteilung in Subnetze an Nicht-Oktettgrenzen« in Kapitel 9 von *DNS und BIND* sowie Abschnitt 6.3.6 im BIND 9 Administrator Reference Manual.

## 2.19 Ihre Dienste leicht auffindbar machen

### Problem

Sie möchten es den Benutzern leicht machen, die Dienste, die Sie anbieten, zu finden.

### Lösung

Geben Sie Ihren Servern »funktionale« Domain-Namen. Zum Beispiel erwarten die meisten Benutzer, dass sie den FTP-Server einer Organisation unter dem Domain-Namen *ftp*.

*domain-name-der-zone* finden können. In den meisten Fällen kann der Domain-Name ein Alias für den kanonischen Namen des Hosts sein, auf dem der Dienst läuft; bei Nameservern und Mail-Servern ist das allerdings nicht möglich.

Weitere übliche funktionale Domain-Namen sind beispielsweise folgende:

*domain-name-der-zone*

Der Domain-Name der Zone besitzt üblicherweise einen oder mehrere A-Records, die auf den Webserver der Organisation verweisen, und einen oder mehrere MX-Records, die Mailern mitteilen, wohin sie Mails liefern sollen, die an die Benutzer in der Organisation adressiert sind.

*imap.domain-name-der-zone*

Ein IMAP-Mail-Server.

*mail.domain-name-der-zone*

Ein SMTP-Mail-Server. Beachten Sie, dass dieser Domain-Name kein Alias sein darf; er muss Eigentümer eines A-Records sein. Des Weiteren muss der Mail-Server sich selbst unter diesem Domain-Namen kennen, um Mail-Schleifen zu vermeiden.

*ns[N].domain-name-der-zone*

Die autorisierten Nameserver für Ihre Zone. Da es oft mehr als einen gibt, verwenden Sie eine Ganzzahl, um zwischen ihnen zu unterscheiden: *ns1*, *ns2* usw. Oder – für unverbesserliche Geeks – *ns0*, *ns1* usw. Beachten Sie, dass diese Domain-Namen *keine* Aliase sein dürfen; sie *müssen* A-Records besitzen.

*ntp.domain-name-der-zone*

Ein NTP-(Network Time Protocol-)Server. Falls Sie mehr als einen haben, unterscheiden Sie sie durch die Verwendung von *ntp1*, *ntp2* usw.

*pop.domain-name-der-zone*

Ein POP-Mail-Server.

*smtp.domain-name-der-zone*

Eine Alternative zu *mail.domain-name-der-zone*. Genau wie *mail.domain-name-der-zone* muss er einen A-Record besitzen.

*www.domain-name-der-zone*

Diese Konvention ist so weit verbreitet, dass sie kaum beschrieben werden muss, aber die meisten Benutzer erwarten, hier die Website einer Organisation zu finden.

## Erläuterung

Ein großer Vorteil bei der Verwendung funktionaler Domain-Namen besteht darin, dass Sie einen Dienst von einem Host auf einen anderen verschieben können, indem Sie einfach den A- oder CNAME-Record für den funktionalen Domain-Namen ändern, ohne gleich die Konfiguration jedes Clients für diesen Dienst zu ändern. Wenn Sie zum Beispiel Ihren NTP-Server von *a.foo.example* auf *b.foo.example* verschieben sollten, könnten Sie den *ntp.foo.example*-CNAME-Record einfach wie folgt ändern:

```
ntp.foo.example.    IN    CNAME    b.foo.example.
```

Vorausgesetzt, Sie hätten Ihre NTP-Clients so konfiguriert, dass sie unter dem Domain-Namen *ntp.foo.example* auf Ihren NTP-Server zugreifen, bräuchten Sie keine Änderungen an der Konfiguration Ihrer Clients vorzunehmen.

Die Domain-Namen von Mail-Servern und Nameservern sind etwas Besonderes bezüglich ihrer Anwendungsweise. Der Domain-Name eines Nameservers erscheint üblicherweise in einem NS-Record und delegiert eine Zone an diesen Nameserver. Ein Nameserver, der diesen NS-Record in einer Antwort sendet, wird in die Antwort lediglich A-Records für den Domain-Namen des Nameservers einfügen. Falls der Domain-Name einen CNAME-Record besitzt, wird der Nameserver ihn nicht finden.

Entsprechend erwarten Mailserver, die Mail an Ihre E-Mail-Adresse senden, A-Records für die Mail-Server, die Sie in Ihren MX-Records auflisten. Wenn Sie CNAME-Records verwenden, finden sie nicht die Adressen, die sie suchen.

Falls darüber hinaus einer Ihrer Backup-Mail-Server die E-Mail empfängt, »stutzt« er die Liste der MX-Records, indem er sich selbst und alle Mail-Server mit niedrigerer Präferenz entfernt. Falls er sich selbst nicht in der Liste findet, weil Sie ein Alias in einem MX-Record verwendet haben, könnte er versuchen, die Mail an sich selbst oder einen weniger bevorzugten Mailserver zu schicken.

## 2.20 Den Standort eines Hosts im DNS ablegen

### Problem

Sie möchten in Ihren Zonendaten den Standort eines Hosts ablegen.

### Lösung

Je nachdem, was Sie mit »Standort« meinen, fügen Sie entweder einen TXT- oder einen LOC-Record für den Domain-Namen des Hosts hinzu.

Viele Administratoren möchten eine beschreibende Ortsangabe für den Host im DNS ablegen. Zum Beispiel könnte es sein, dass Sie angeben möchten, dass sich der Host *a.foo.example* in Ihrem Gebäude 20 auf Ebene C in der Nähe von Position C3K befindet. Dazu könnten Sie den folgenden TXT-Record zu Ihrer Zone hinzufügen:

```
a.foo.example.    IN    TXT    "Building 20, level C, post C3K"
```

Falls Sie andererseits eine geografische Ortsangabe für den Host angeben möchten (d. h. seinen Breitengrad, seinen Längengrad und seine Höhe über NN), können Sie einen LOC-Record zu Ihrer Zone hinzufügen. Falls sich *a.foo.example* auch bei 40 Grad, 2 Minuten, 0,373 Sekunden nördlicher Breite, 105 Grad, 17 Minuten, 23,528 Sekunden westlicher Länge und in 1.638 Metern Höhe befindet, könnten Sie den folgenden LOC-Record zu Ihrer Zone hinzufügen:

```
a.foo.example.    IN      LOC      40 2 0.373 N 105 17 23.528 W 1638m
```

## Erläuterung

Der TXT-Record ist bemerkenswert vielseitig, da Sie wirklich fast *alles* in die RDATA-Felder schreiben können. Denken Sie lediglich daran, dass Leute, die wissen, wie man TXT-Records für einen Domain-Namen nachschlägt, die Daten finden werden, die Sie dort ablegen. Falls Sie darüber hinaus mehrere TXT-Records für einen Domain-Namen hinzufügen, gibt es keine Garantie, in welcher Reihenfolge der Nameserver sie zurückgeben wird.

Der LOC-Record dagegen ist absolut spezialisiert: Er speichert lediglich geografische Ortsdaten. Das Format ist genau so, wie ich es weiter oben gezeigt habe, mit separaten RDATA-Feldern für Grad, Minuten und Sekunden, gefolgt von N für Nord, S für Süd, E für Ost (bzw. East) und W für West. Und Sie können negative Höhenwerte verwenden, falls Sie zufälligerweise in einem Bergwerk im Death Valley sitzen.

Falls Sie nicht sicher sind, wie der Breitengrad, der Längengrad und die Höhe Ihres Standorts lauten und Sie Ihren Chef nicht überzeugen können, dass Sie einen GPS-Empfänger brauchen, um es herauszufinden, können Sie den Eagle Geocoder von Etak ([www.geocode.com/eagle.html-ssi](http://www.geocode.com/eagle.html-ssi)) oder die Airport Information von AirNav ([www.airnav.com/airports/](http://www.airnav.com/airports/)) verwenden, um die Daten für Ihre Adresse beziehungsweise für den nächstgelegenen Flughafen zu ermitteln.

## Siehe auch

Für weitere Informationen über LOC-Records siehe den Abschnitt »Standort« in Kapitel 16 von *DNS und BIND*, RFC 1876 oder die hervorragende Website von Christopher Davis unter <http://www.ckdhr.com/dns-loc/>.

## 2.21 Eine Host-Tabelle filtern, um Zonendaten-Dateien zu erhalten

### Problem

Sie möchten eine existierende Host-Tabelle, etwa eine */etc/hosts*-Datei, filtern, um Zonendaten-Dateien zu erhalten.

### Lösung

Verwenden Sie ein Tool wie etwa *h2n*, um Ihre Host-Tabelle in die entsprechenden Zonendaten-Dateien umzuwandeln. Bei *h2n* geben Sie den Domain-Namen der zu erzeugenden Forward-Mapping-Zone als Argument der Option *-d* an und Sie geben die mit

dieser Zone verknüpften Netzwerke als Argument einer oder mehrerer `-n`-Optionen an. Zum Beispiel erzeugt der folgende Befehl Datendateien für die Zonen *foo.example* und *168.192.in-addr.arpa*:

```
% h2n -d foo.example -n 192.168
```

Diese Zonendaten-Dateien würden jeweils einen SOA-Record und einen NS-Record enthalten, die auf den lokalen Host verweisen, außerdem A-Records oder PTR-Records für die Hosts aus dem Netzwerk 192.168/16 in */etc/hosts*. Zusätzliche Optionen ermöglichen es Ihnen, andere Records zu erzeugen, darunter NS-Records, die auf andere Nameserver verweisen.

## Erläuterung

Sie erhalten eine Kopie von *h2n* mit dem Tar-Archiv, das zu *DNS und BIND* gehört und sich unter [ftp.oreilly.com/published/oreilly/nutshell/dnsbind/dns.tar.Z](http://ftp.oreilly.com/published/oreilly/nutshell/dnsbind/dns.tar.Z) befindet. Außerdem hat Andris Kalnozols von Hewlett-Packard *h2n* bedeutend verbessert; er stellt seine verbesserte Version unter [ftp.hpl.hp.com/pub/h2n/h2n.tar.gz](http://ftp.hpl.hp.com/pub/h2n/h2n.tar.gz) zur Verfügung.

Es gibt noch andere Tools, die Daten im Host-Tabellen-Format in Zonendaten-Dateien umwandeln; *h2n* ist nur eine Option. Schauen Sie sich für einige verfügbare Optionen den Inhalt von *bind-contrib.tar.gz* an, das Sie im gleichen Verzeichnis wie das neueste BIND 8-Release finden.

## Siehe auch

Rezept 1.11 zum Beschaffen einer Kopie von BIND (oder *bind-contrib.tar.gz*) und »Werkzeuge« in Kapitel 4 von *DNS und BIND*.