

D *Das Programm ^asendmail^o*

Aus der Sicht des Benutzers ist der Versand einer Mail-Mitteilung eine einfache Prozedur. Wie Sie jedoch wissen oder vermuten, kann die Zustellung einer Mitteilung an ihren Empfänger eine recht komplizierte Sache sein. Ein wesentlicher Bestandteil des Mail-Systems ist das Dienstprogramm **sendmail**, ein Leitweglenkungsprogramm für Mail-Mitteilungen. Die Post wird von einem Postversandprogramm auf Benutzerebene, beispielsweise NeXTmail, eingesammelt und an Postweiterleitungs- oder zustellungsprogramme, wie z. B. UUCP, verteilt.

Um **sendmail** zu verstehen, müssen Sie über die **sendmail**-Konfigurationsdateien Bescheid wissen und über deren Beziehung zur Umgebung, in der sie interpretiert werden (eventuell umfaßt diese NetInfo, NIS und DNS). In diesem Anhang erhalten Sie Hintergrundinformationen über die Funktionsweise von **sendmail**, Erläuterungen zu den Konfigurationsdateien und Beispiele zu Debugläufen. Das Dienstprogramm **sendmail** ist für seine Komplexität bekannt. Dieser Anhang soll Ihnen die Grundlagen vermitteln und ist nicht als komplette Beschreibung gedacht. Im Anhang H, ^aFachliteratur-Hinweise^o, finden Sie eine Liste mit Dokumentationen über **sendmail**.

Hinweis: Die Angaben in diesem Anhang, insbesondere die Tabellen, erfolgen in Anlehnung an Eric Allmans ^aSendmail Installation and Operation Guide^o. Dieser Text ist im *UNIX System Manager's Manual, 4.3 Berkeley Software Distribution* enthalten. (Einzelheiten hierzu finden Sie in Anhang H).

Warnung: Das **sendmail**-System dient der allgemeinen Postadressierung und -weiterleitung. Diese Aufgaben sind äußerst kompliziert. Modifizieren Sie die vorhandenen Konfigurationsdateien nur, wenn Sie absolut sicher sind, daß dies notwendig ist, und wenn Sie sich die damit verbundene Arbeit zutrauen. Erstellen Sie unbedingt Sicherungskopien der vorhandenen Dateien, falls etwas schief geht.

Hintergrund

Das Programm **sendmail** erledigt vier Aufgaben:

- Das Programm bestimmt, wie eine Mitteilung bearbeitet werden soll & d. h., wie sie an ein Postversandprogramm weitergeleitet wird.
- Die Kopfzeile der Mitteilung wird teilweise umgeschrieben, um den Adressen das richtige Format und den richtigen Inhalt zu geben.
- Das Programm führt Alias-Auflösungen durch (dabei werden Aliase in Benutzernamen umgesetzt) und verarbeitet **.forward**-Dateien für die lokale Zustellung.
- Die Post wird mit SMTP weitergeleitet (Simple Mail Transfer Protocol, das TCP/IP-Netzwerkprotokoll für Postübertragung).

Beachten Sie, daß **sendmail** die Post nicht zustellt und auch nicht weiterleitet & mit Ausnahme der Mail-Mitteilungen, die über SMTP weitergeleitet wurden. **sendmail** benötigt andere Programme (zum Beispiel **/bin/mail** oder **uux**), um Mail-Mitteilungen zuzustellen und weiterzuleiten.

Das Programm **sendmail** regelt den Postverkehr. Die Mitteilungen werden von den Programmen übernommen, die Post versenden & wie z. B. NeXTmail. Anschließend bestimmt **sendmail**, wie die Mitteilungen weitergeleitet werden sollen und übergibt sie einem Postweiterleitungs- oder zustellungsmedium. Das Weiterleitungsmedium kann z. B. **sendmail** auf einem anderen Computer in Verbindung mit SMTP sein oder auch das Programm **rmail** eines anderen Computers in Verbindung mit UUCP. Als Zustellungsmedium für die lokale Zustellung kann beispielsweise **/bin/mail** verwendet werden. Die Anweisungen zur Bearbeitung einer bestimmten Mitteilung befinden sich in der **sendmail**-Konfigurationsdatei.

Konfigurationsdateien

Zusätzlich zu den Bearbeitungsanweisungen für eine bestimmte Mail-Mitteilung enthält die Konfigurationsdatei auch spezifische Informationen über das Netzwerk, auf dem **sendmail** ausgeführt wird. Jeder Computer kann zwar seine eigene Konfigurationsdatei haben, im Normalfall sind die Konfigurationsdateien der meisten Computer eines bestimmten Netzwerkes aber identisch. In der Regel unterscheidet sich die Konfigurationsdatei des Mail-Servers von denen der Mail-Clients. Die Dateien der Clients sind jedoch identisch.

Die voreingestellte **sendmail**-Konfigurationsdatei ist **/etc/sendmail/sendmail.cf**. Dies könnte ein symbolischer Verweis auf eine andere Datei in **/etc/sendmail** sein, wie etwa **sendmail.mailhost.cf**. Zusätzlich kann die Lage der Konfigurationsdatei als Wert der Eigenschaft **sendmail.cf** im NetInfo-Verzeichnis **/locations/sendmail** in der NetInfo-Datenbank gespeichert werden. Weitere Informationen hierzu finden Sie im Kapitel 6, ^aSo verwalten Sie die elektronische Post^o.

Hinweis: Einige Ausführungen von **sendmail** unterstützen eine "eingefrorene" Konfigurationsdatei, nämlich **sendmail.fc**. Diese Datei wird von NeXT-Computern nicht unterstützt. Die Möglichkeit, solch eine Konfigurationsdatei zu erstellen, wurde entfernt.

Kopfzeilen und Umschläge

Mail-Mitteilungen enthalten die Adresse an zwei Stellen: in der *Kopfzeile* der Mitteilung (beispielsweise in den Zeilen **From:**, **To:** und **Cc:**) und auf dem *Umschlag* der Mitteilung. Der Umschlag enthält die Rückadresse und alle Zieladressen. Die Mitteilungen werden den Zieladressen zugestellt, die auf dem Umschlag angegeben sind. Die Umschlagadressen sind in der Befehlszeile angegeben. Wenn Sie NeXTmail verwenden, sind diese nicht sichtbar. Einzelheiten hierzu finden Sie weiter hinten in diesem Kapitel unter "Branchenkniffe".

Die Empfänger sehen nicht alle Umschlagadressen (das ist der springende Punkt bei einer "Blindkopie"). Die Umschlagrückadresse finden Sie in den Kopfzeilen **From** (nicht zu verwechseln mit **From:**) und **Return-Path:** der zugestellten Mitteilung.

Die Komponenten der Konfigurationsdatei

Die **sendmail**-Konfigurationsdatei besteht aus fünf Grundkomponenten. Bedeutung und Komplexität nehmen in der Reihenfolge zu: Optionen, Makros, Klassen, Mail-Programme sowie Regelsätze und Umschreiberegeln.

Warnung: Die Modifikation der **sendmail**-Konfigurationsdateien sollten Sie nur als letzte Lösung erwägen. Versuchen Sie erst, eine vorhandene Konfigurationsdatei zu finden, die Ihren Bedürfnissen entspricht (oder einen Satz solcher Dateien). Wenn Sie die **sendmail**-Konfigurationsdatei modifizieren, sollten Sie erst eine Sicherungskopie anlegen, für den Fall, daß die Modifikationen nicht Ihren Erwartungen entsprechen. Es ist *überaus* einfach, Ihr Mail-System mit der mißlungenen Modifikation einer **sendmail**-Konfigurationsdatei außer Betrieb zu setzen.

Optionen

Optionen steuern verschiedene einfache Aspekte von **sendmail**. Sie können Optionen sowohl in der Konfigurationsdatei als auch in der Befehlszeile setzen, mit der **sendmail** aufgerufen wird (normalerweise von **/etc/rc** ausgeführt).

In der Konfigurationsdatei werden Optionen mit der Anweisung **O** gesetzt. Nachfolgend ein Beispiel:

```
OT3d
```

Diese Option legt fest, daß die Post zurückgesendet wird, die innerhalb von 3 Tagen nicht zugestellt wurde. Andere Optionen steuern diverse Aspekte wie Berechtigungen ("Modus") temporärer Dateien, Lagen der Aliase-Dateien und des Spool-Verzeichnisses sowie die SMTP-Zeitüberschreitung. Einzelheiten hierzu finden Sie im UNIX-Handbuch unter **sendmail**.

Makros

Makros sind einfache Variablen. Manche Makros werden von **sendmail** selbst bestimmt, z. B. das aktuelle Datum und der Hostname des Absenders. Andere können in der **sendmail**-Konfigurationsdatei definiert werden. Solche Makros könnten beispielsweise den Namen eines Computers enthalten, an den die Post gesendet werden soll, die lokal nicht zugestellt werden kann. Makronamen bestehen aus einem einzigen Buchstaben.

Zahlreiche integrierte Makros sind verfügbar. Sie sind alle mit einem Kleinbuchstaben benannt. Normalerweise sind Großbuchstaben für Makros reserviert, die vom Benutzer definiert wurden.

Makros werden mit der Anweisung **D** definiert:

```
DMetherl
```

In diesem Beispiel wird bestimmt, daß das Makro **M** die Zeichenfolge **etherl** enthält.

Klassen

Klassen dienen dazu, gleichartige Elemente zusammenzufassen. Normalerweise zeigt eine Klasse an, daß eine Gruppe von Hosts oder Hostnamen in der gleichen Weise behandelt werden soll. Alle Computer, die UUCP-Partner sind, könnten z. B. in einer Klasse zusammengefaßt werden, oder auch alle verschiedenen Namen des aktuellen Computers.

Klassen werden mit den Anweisungen **C** und **F** definiert. Mit **C** listen Sie die Mitglieder der Klasse ausdrücklich auf; mit **F** werden die Mitglieder aus einer Datei oder von der Standardausgabe eines Befehls geholt. Nachstehend je ein Beispiel:

```
CR mailhost mail-relay mail-server
```

```
FV|/usr/bin/uuname  
FA/etc/sysadmins
```

In der ersten Zeile wird die Klasse **R** so definiert, daß sie die Zeichenfolgen **mailhost**, **mail-relay** und **mail-server** enthält. Die zweite Zeile definiert die Klasse **V** so, daß sie die Ausgabe des Befehls **/usr/bin/uuname** enthält. Im dritten Beispiel wird schließlich die Klasse **A** definiert & sie soll den Inhalt der Datei **/etc/sysadmins** enthalten.

Mail-Programme

Ein Mail-Programm definiert die Programme und Optionen, mit denen Post weitergeleitet oder zugestellt wird. Verschiedene Mail-Programme werden u. a. aus folgenden Gründen verwendet:

- Die Mitteilungen werden über unterschiedliche Datenträger weitergeleitet, oder es werden verschiedene Protokolle verwendet (z. B. UUCP und SMTP).
- Verschiedene Optionen (z. B. maximale Länge der Mitteilung) werden unter verschiedenen Bedingungen verwendet.
- Verschiedene Methoden werden verwendet, um die Adressen des Absenders und der Empfänger umzuschreiben.

Mail-Programme können beliebig benannt werden. Eine Ausnahme bilden drei Mail- Programme, die bei **sendmail** eine bestimmte Bedeutung haben: **local**, **prog** und **error**.

Mail-Programme werden mit der Anweisung **M** definiert:

```
Muucp, P=/usr/bin/uux, F=msDFMhuU, S=13, R=23, E=\n  
A=uux - -r $h!rmail ($u)
```

In diesem Beispiel wird das Mail-Programm **uucp** definiert. Nachfolgend eine Aufschlüsselung der verschiedenen Komponenten:

- **P** (Pfad) & Bestimmt das Programm, das aufgerufen werden soll. In diesem Fall ist dies **/usr/bin/uux**.
- **F** (Flaggen) & Bestimmt die *Mail-Programm-Flaggen*, mit denen sowohl die Einzelheiten der Ausführung als auch die Argumente des Programms bestimmt werden, das für dieses Mail-Programm aufgerufen wurde. Eine vollständige Beschreibung finden Sie in der Dokumentation von Eric Allman.
- **S** (Sender, Absender) & Bestimmt den Regelsatz, mit dem die Absenderadressen der Mitteilungen umgeschrieben werden, die an dieses Mail-Programm gesendet wurden. Weitere Informationen hierzu finden Sie weiter hinten in diesem Kapitel unter *Fluß und Anwendung der Regeln*.

- **R** (Recipient, Empfänger) ⚭ Genau wie **S**, jedoch für die Empfängeradressen.
- **E** (eol) ⚭ Bestimmt die Zeichenfolge für das Zeilenende (end-of-line, eol). In diesem Fall ist dies **\n** (Zeichen für eine neue Zeile).
- **A** (argv) ⚭ Bestimmt den Argumentvektor, der an das Programm gesendet wird. Zunächst wird der Programmname wiederholt (häufig in verkürzter Form), gefolgt von den Argumenten für das Programm, die Sie in die Befehlszeile eingeben würden.

Im obigen Beispiel wurde die Option **M** (maxsize) nicht verwendet. Sie bestimmt die maximal zulässige Länge einer Mitteilung (in Bytes). Falls diese Länge überschritten wird, kann die Mitteilung nicht zugestellt werden, und der Absender erhält eine Fehlermeldung.

Das Mail-Programm **local**

Das Mail-Programm **local** muß in der Konfigurationsdatei definiert werden. Dieses Mail-Programm wird für die Endzustellung einer Mitteilung an Benutzer auf dem lokalen Computer verwendet. Alias-Auflösungen werden nur bei Mail-Mitteilungen durchgeführt, die vom Mail-Programm **local** verarbeitet werden.

Das Mail-Programm **local** ruft normalerweise **/bin/mail** auf. Dieses Programm schreibt eine Kopie der Mitteilung in die Mail-Spool-Datei des Benutzers:

```
Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d $u
```

Das Mail-Programm **prog**

Das Mail-Programm **prog** muß ebenfalls in der Konfigurationsdatei definiert werden. Dieses Mail-Programm wird für die Endzustellung einer Mitteilung verwendet, wenn als Empfängeradresse ein Programm angegeben wurde. Vom folgenden Mail-Alias wird beispielsweise das Programm **/usr/local/bin/ambigmail** als Empfänger der eingehenden Mitteilung angegeben:

```
|"/usr/local/bin/ambigmail"
```

Für solche Adressen wird das Mail-Programm **prog** anstelle des Mail-Programms **local** aufgerufen. Das Mail-Programm **prog** ruft normalerweise **/bin/sh** auf:

```
Mprog, P=/bin/sh, F=lsDFMeup, S=10, R=20, A=sh -c $u
```

Das Mail-Programm **error**

Das Mail-Programm **error** ist in **sendmail** integriert und sollte nicht in der Konfigurationsdatei definiert werden. Dieses Mail-Programm erstattet Fehlerberichte über unbekannte lokale Hosts an den Absender und wahrscheinlich auch an **postmaster**. Dabei handelt es sich um ein Alias für die Benutzer, die für die Mail-Installation zuständig sind. Ein Beispiel für die Verwendung des Mail-Programms **error** finden Sie im nächsten Abschnitt.

Fehlermeldungen können auch von anderen Quellen generiert werden. Wenn das Mail-Programm **local** beispielsweise feststellt, daß ein Benutzer nicht lokal existiert, wird eine Fehlermeldung generiert. In diesem Fall wird nicht das Mail-Programm **error** für den Fehlerbericht verwendet.

Regelsätze und Umschreiberegeln

Der Großteil einer **sendmail.cf**-Datei besteht aus Anweisungen dazu, wie Mail-Adressen interpretiert und modifiziert werden. Diese Anweisungen werden *Umschreiberegeln* genannt und in *Regelsätzen* zusammengefaßt. In einem Regelsatz werden verschiedene Regeln zusammengefaßt, die eine bestimmte Funktion genau definieren. Ein Regelsatz wandelt Adressen beispielsweise in ein standardmäßiges (*kanonisches*) Format um, das von den anderen Regelsätzen einfacher interpretiert werden kann.

Regelsätze werden mit einer Zahl benannt und mit der Anweisung **S** definiert:

```
S30
```

Auf einen Regelsatz folgen sofort die Umschreiberegeln, aus denen dieser Regelsatz besteht. Umschreiberegeln werden mit der Anweisung **R** definiert:

```
R$+@$+      $:$1<@$2>      focus on domain
```

Diese Regel besteht aus drei Komponenten, die durch Tabulatorzeichen voneinander getrennt werden: die linke Seite, die rechte Seite und ein Kommentar.

Die linke Seite

Das Muster der linken Seite einer Regel kann mit einer Adresse verglichen werden. Es gleicht einem regulären Ausdruck, abgesehen davon, daß es *Zeichenfolgen* oder Wörter vergleicht und also nicht einzelne Zeichen. Folgende Adresse hat z. B. fünf Zeichenfolgen:

```
amm@Tute.EDU
```

Die fünf Zeichenfolgen sind **amm**, **@**, **Tute**, **.**, und **EDU**.

In der folgenden Tabelle werden die Metasymbole beschrieben, die in der Kombination auf der linken Seite verwendet werden:

| Metasymbol | Bedeutung |
|------------|---|
| \$* | Null oder mehreren Zeichenfolgen entsprechen. |
| \$+ | Einer oder mehreren Zeichenfolgen entsprechen. |
| \$- | Genau einer Zeichenfolge entsprechen. |
| \$=c | Irgendeiner Zeichenfolge der Klasse <i>c</i> entsprechen. |
| \$~c | Irgendeiner Zeichenfolge außer der Klasse <i>c</i> entsprechen. |

Nachfolgend wird das Beispiel der Umschreibregel noch einmal aufgenommen:

`R$+@$+$` `$. $1<@$2>` `focus on domain`

Die linke Seite dieser Regel entspricht einer oder mehreren Zeichenfolgen (**\$+**), gefolgt von der Zeichenfolge **@** und einer oder mehreren anderen Zeichenfolgen (**\$+**).

Die rechte Seite

Auf der rechten Seite einer Regel wird angegeben, was mit einer Adresse geschehen soll, wenn sie der linken Seite der Regel entspricht. In der folgenden Tabelle werden die Metasymbole beschrieben, die auf der rechten Seite einer Umschreibregel verwendet werden:

| Metasymbol | Bedeutung |
|-----------------------|--|
| \$ <i>n</i> | Die Zeichenfolgen ersetzen, die Metasymbol-Nr. <i>n</i> der linken Seite entsprechen. |
| \$(<i>name</i> \$) | Den Hostnamen (oder die Hostadresse) <i>name</i> in Standardformat (<i>kanonisch</i>) umwandeln. |
| \$> <i>n</i> | Regelsatz-Nr. <i>n</i> aufrufen. |
| \$# <i>name</i> | Die Adresse für das Mail-Programm <i>name</i> auflösen. |
| \$@ <i>name</i> | Den Host als <i>name</i> angeben. |
| \$. <i>name</i> | Den Benutzer als <i>name</i> angeben. |
| \$(<i>m name</i> \$) | <i>name</i> in der NIS-Tabelle nachschlagen, die im Makro <i>m</i> angegeben ist. |

Zusätzlich zu diesen Metasymbolen können Sie die Wirkung der Regel oder der Regelsätze auch mit zwei Präfixen modifizieren. Dies wird in folgender Tabelle beschrieben:

| Präfix | Bedeutung |
|--------|--|
| \$: | Diese Regel sofort beenden, mit dem Regelsatz jedoch fortfahren. |

\$@

Von diesem Regelsatz mit dem Rest der rechten Seite als Resultat des Regelsatzes zurückkehren.

Nachfolgend noch einmal das Beispiel der Umschreibregel:

```
R$+@$+      $:$1<@$2>      focus on domain
```

Die rechte Seite schreibt die Adresse in folgendes Format um:

```
Zeichenfolge1<@Zeichenfolge2>
```

Würde diese Regel auf die Adresse **amm@Tute.EDU** angewendet, entspräche *Zeichenfolge1* **amm**, und *Zeichenfolge2* entspräche **Tute.EDU**. Die rechte Seite der Regel wandelt die Adresse folgendermaßen um:

```
amm<@Tute.EDU>
```

Der Domainteil der Adresse wird in spitze Klammern (<>) eingeschlossen und ist so für andere Regeln einfacher zu identifizieren. (Dies wird *Konzentrierung* genannt).

Durch das rechte Präfix **\$:** wird die Regel beendet. Dadurch wird vermieden, daß die Regel auf die umgeschriebene Adresse angewendet wird. (Die umgeschriebene Adresse entspricht eventuell immer noch der linken Seite, die sonst verursachen würde, daß die Regel auf die umgeschriebene Adresse angewendet wird.)

Nachfolgend ein Beispiel für eine Umschreibregel, die das Mail-Programm **error** aufruft:

```
R$*<@$*.LOCAL>$*      $#error $:Never heard of host $2 in domain $m
```

Wenn die linke Seite der Adresse entspricht, wird das Mail-Programm **error** aufgerufen, um die angegebene Fehlermeldung zurückzugeben (alles hinter **\$:**). Tabulatorzeichen können im Text einer Fehlermeldung nicht verwendet werden, da dieses Zeichen die rechte Seite einer Umschreibregel vom Kommentar trennt.

Hinweis: Das Mail-Programm **error** sollte nur im Regelsatz 0 aufgerufen werden. (Einzelheiten finden Sie im nächsten Abschnitt).

So werden Adressen umgesetzt

Der wichtigste Aspekt der **sendmail**-Konfigurationsdateien liegt vielleicht darin, wie Regelsätze zur Umsetzung einer Mail-Adresse verwendet werden.

Adressensyntax

E-Mail-Adressen haben zahlreiche verschiedene Formate & deshalb sind **sendmail**-Konfigurationsdateien auch so au  erordentlich komplex.

Im allgemeinen hat eine Mail-Adresse eines der folgenden drei Formate:

- Einfacher Benutzername oder Alias & wie etwa **suser**, **Sandy_User** oder **ges_Abt**.
- Benutzername oder Alias auf einem anderen Computer & wie etwa **suser@Rhino.COM**.
- Benutzername oder Alias auf einem anderen Computer, der   ber UUCP erreichbar ist & wie etwa **rhino!suser**.

Andere Adressenformate sind m  glich, einschlie  lich kombinierter Formate. In der folgenden Tabelle werden einige Beispiele f  r Mail-Adressen beschrieben:

| Adresse | Interpretation |
|-------------------------|---|
| amm | Lokaler Benutzer oder Alias amm |
| amm@rhino | amm auf dem Computer rhino in der lokalen Domain |
| rhino!amm | amm auf dem Computer rhino ,   ber UUCP erreichbar |
| amm@rhino.UUCP | amm auf dem Computer rhino ,   ber UUCP erreichbar |
| rhino!hippo!conure!amm | amm auf dem Computer conure ,   ber UUCP und die Computer rhino und hippo erreichbar |
| amm@Rhino.COM | amm in der Domain Rhino.COM |
| amm@flyer.Rhino.COM | amm in der Domain flyer.Rhino.COM ; normalerweise ist dies der Computer flyer in der Domain Rhino.COM |
| @Rhino.COM:amm@Tute.EDU | amm in der Domain Tute.EDU , erreichbar   ber die Domain Rhino.COM |
| amm%rhino.UUCP@Tute.EDU | amm auf dem Computer rhino.UUCP , erreichbar   ber die Domain Tute.EDU |
| amm%Rhino.COM@Tute.EDU | amm in der Domain Rhino.COM , erreichbar   ber Tute.EDU |

Die **sendmail**-Konfigurationsdatei setzt alle Adressen in konforme, sogenannte *Domainformat*-Adressen um. Eine einfache Domainformat-Adresse w  re *benutzer@domain*, wobei *domain* der Name eines Computers oder einer Computergruppe ist (z. B. eines Netzwerkes). Wenn *domain* eine Computergruppe ist, wird ein Computer f  r die Bearbeitung der eingehenden Post dieser Gruppe bereitgestellt. Das Programm **sendmail** (oder ein entsprechendes Programm) auf dem Computer, der die Post f  r *domain* bearbeitet, interpretiert *benutzer*.

Die Adresse **rhino!suser** ist z. B. keine Domainformat-Adresse und wird deshalb in **suser@rhino.uucp** umgesetzt. In diesem Fall ist **rhino.uucp** lediglich eine zweckm   ige Notation, denn es gibt keine Domain **uucp**

und somit auch keine untergeordnete Domain **rhino.uucp**. Die Umsetzung in eine standardmäßige, kanonische Domainformat-Adresse wird *Kanonisation* genannt.

Im allgemeinen werden Mail-Adressen nicht nach Groß- und Kleinschreibung unterschieden: Die Adressen **amm@rhino** und **amm@Rhino** sind die gleichen. UUCP-Adressen bilden jedoch eine Ausnahme: **amm@rhino.uucp** ist *nicht* dasselbe wie **amm@Rhino.uucp**.

Fluss und Anwendung der Regelsätze

Regelsätze werden in einer bestimmten Reihenfolge auf Adressen angewendet. In diesem Abschnitt wird beschrieben, welche Regelsätze auf die verschiedenen Adressenarten angewendet werden. Die Regelsätze selbst werden im nächsten Abschnitt beschrieben.

Die folgende Abbildung zeigt die Regelsätze, die während der Mail-Programm-Auflösung verwendet werden & mit denen also bestimmt wird, was mit einer Mitteilung geschehen soll, die an eine bestimmte Adresse gesendet wird:

F0.eps ,

Die Umschlagadresse (die sich von der Adresse in der Mitteilung unterscheiden kann oder eventuell dort auch gar nicht erscheint) wird zunächst vom Regelsatz 3 und danach vom Regelsatz 0 verarbeitet. Regelsatz 3 hat die Funktion, die Adresse in ein standardmäßiges (kanonisches) Format umzusetzen und **sendmail** auf den Domainteil der Adresse zu "konzentrieren", d. h. in spitze Klammern (<>) zu setzen. Regelsatz 0 bestimmt, was mit der Mitteilung geschehen soll, die an diese Adresse gesendet wird.

Die nächste Abbildung zeigt die Regelsätze, mit denen die Absenderadresse in der Kopfzeile der Mitteilung (etwa in der Zeile **From:**) umgeschrieben wird.

F1.eps ,

Regelsatz 3 wird zuerst aufgerufen, gefolgt von einem Prozess, der *Hinzufügen der Absender-Domain* genannt wird (falls angegeben). Dadurch wird der Adresse bei Bedarf ein Domainteil hinzugefügt. Anschließend wird der Regelsatz 1 aufgerufen, der eine allgemeine Vorverarbeitung der Absenderadresse vornimmt & beispielsweise wird ein "login"-Name in ein Alias umgesetzt. Danach wird der Regelsatz aufgerufen, der von der Option **S** der Mail-Programm-Definition angegeben wird. Schließlich folgt Regelsatz 4, der eine allgemeine Nachbearbeitung vornimmt & u. a. wird die Konzentrierung teilweise rückgängig gemacht, die von Regelsatz 3 durchgeführt wurde.

Die nächste Abbildung zeigt die Regelsätze, mit denen Empfängeradressen in der Kopfzeile der Mitteilung (wie etwa in der Zeile **To:**) umgeschrieben werden.

F2.eps ,

Regelsatz 3 und Regelsatz D sind hier mit dem vorigen Beispiel identisch. Regelsatz 2 führt eine allgemeine Verarbeitung der Empfängeradresse durch (häufig nicht erforderlich, deshalb ist der Regelsatz 2 oft leer). Danach wird der Regelsatz aufgerufen, der von der Option **R** der Mail-Programm-Definition bestimmt wird. Schließlich folgt Regelsatz 4, der eine allgemeine Nachverarbeitung durchführt.

Die folgende Abbildung kombiniert die vorherigen drei Abbildungen und zeigt, wie die verschiedenen Regelsätze bei der Verarbeitung einer Adresse verwendet werden. Auf welchem Pfad eine Adresse durch die Regelsätze geschleust wird, hängt von ihrem Typ ab. Das Diagramm wurde aus dem erwähnten Text von Eric Allman übernommen und erscheint in vielen Dokumentationen über **sendmail**.

F3.eps ,

Spezifische Regelsätze

Nachfolgend finden Sie eine genaue Beschreibung der verschiedenen Regelsätze, die im vorherigen Abschnitt erwähnt wurden.

Regelsatz 3 ➤ Kanonisch umsetzen und konzentrieren

Regelsatz 3 setzt eine Adresse in ein standardmäßiges Format um (Kanonisation) und schließt den Domainteil der Adresse in spitze Klammern ein (Konzentrierung). Die weiter vorne verwendete Umschreibregel ist ein Beispiel für Regelsatz 3.

Nachfolgend einige Beispiele, wie Adressen vom Regelsatz 3 umgesetzt werden:

| Originaladresse | Nach der Umsetzung |
|-----------------|--------------------|
| amm@Rhino.COM | amm<@Rhino.COM> |
| rhino!amm | amm<@rhino.uucp> |

Regelsatz 0 ➤ Mail-Programm-Auflösung

Regelsatz 0 bestimmt, was mit der Mitteilung an eine bestimmte Adresse geschehen soll: Mit welchem Mail-

Programm soll die Post weitergeleitet oder zugestellt werden; welcher Host empfängt die Mitteilung, wenn sie weitergeleitet werden soll, und welche Adresse soll dem Weiterleitungs- oder Zustellungsmedium gegeben werden. Dieser Prozeß wird *Auflösung* der Adresse genannt. Die Ausgabe des Regelsatzes 0 enthält drei Elemente:

- das Mail-Programm, mit dem die Post weitergeleitet oder zugestellt wird (z. B. **ddn**);
- den Host, an den die Mitteilung gesendet werden soll (z. B. **Tute.EDU**);
- die Adresse des Benutzers, der die Mitteilung empfangen soll (z. B. **amm@Tute.EDU**).

Wenn die Post lokal zugestellt werden soll, enthält die Ausgabe keine Daten über den Host.

Die nächste Tabelle zeigt das Resultat, nachdem der Regelsatz 0 auf mehrere Adressen angewandt wurde. In dieser Tabelle wird vorausgesetzt, daß die Adressenauflösung auf dem Mail-Server durchgeführt wird. (Dabei handelt es sich um den Computer, der für die Zustellung lokaler Post und die Weiterleitung entfernter Post zuständig ist.) Weiterhin wird vorausgesetzt, daß Post an Computer anderer Anlagen über das Internet zugestellt wird.

| Adresse | Mail-programm | Host | Benutzer |
|--------------------------|---------------|----------|--------------------------|
| suser | local | | suser |
| suser@rhino | ether | rhino | suser@rhino |
| suser@Tute.EDU | ddn | Tute.EDU | suser@Tute.EDU |
| rhino!suser | uucp | rhino | suser |
| suser@rhino.uucp | uucp | rhino | suser |
| rhino!conure!suser | uucp | rhino | conure!suser |
| suser%Rhino.COM@Tute.EDU | ddn | Tute.EDU | suser%Rhino.COM@Tute.EDU |

Möchten Sie zur Weiterleitung von Post, die an Computer anderer Anlagen adressiert ist, ein anderes Mail-Programm verwenden, müssen Sie dieses mit der Anweisung **M** definieren. Dies wurde weiter vorne beschrieben. Außerdem sollte das Makro **M** als Name dieses Mail-Programms definiert werden. Das Makro **R** sollte als bevorzugter Name des entfernten Weiterleitungs-Computers definiert werden. Die Klasse **R** sollte den bevorzugten Namen sowie alle Aliase des entfernten Weiterleitungs-Computers enthalten.

Regelsatz D ๑ Hinzufügen der Absender-Domain

Der Regelsatz D existiert nicht wirklich, es handelt sich vielmehr um eine integrierte Prozedur, genannt ^aHinzufügen der Absender-Domain^o. Die Absenderadresse durchläuft zunächst die Auflösungssequenz des Mail-Programms (Regelsatz 3, dann Regelsatz 0). Danach prüft **sendmail**, ob die Mail-Programm-Flagge **C** für das Mail-Programm vorhanden ist, von dem die Mitteilung *erhalten* wurde. Ist die Flagge vorhanden, wird der Domainteil der Absenderadresse allen Adressen in der Kopfzeile hinzugefügt, die nicht bereits einen Domainteil

haben (die also anscheinend lokale Adressen sind). Damit wird sichergestellt, daß Antworten auf die Mitteilung der richtigen Adresse zugestellt werden. Wenn die Mitteilung zum Beispiel vom Benutzer **amm** gesendet wird, könnte die Funktion ^aHinzufügen der Absender-Domain^o die Adresse in **amm@Rhino.COM** umwandeln.

Regelsatz 1 ➤ Standardmäßige Vorverarbeitung der Absenderadresse

Regelsatz 1 ermöglicht die Verarbeitung einer Absenderadresse für die gesamte Anlage. Eine Anlage könnte z. B. ein Standardformat für Mail-Adressen integrieren, so daß unabhängig vom tatsächlichen ^alogin^o-Namen eines Benutzers die Post dieses Benutzers so erscheint, als käme sie von dieser Standardadresse. Eventuell möchten Sie z. B. einen ^alogin^o-Namen wie **rkabir** in **Randy_Kabir** umwandeln.

Regelsatz 2 ➤ Standardmäßige Vorverarbeitung der Empfängeradresse

Sie können die Empfängeradresse mit Regelsatz 2 genauso standardmäßig vorverarbeiten, wie Sie die Absenderadresse mit Regelsatz 1 standardmäßig vorverarbeiten. Sie könnten beispielsweise die Hostnamen für einige Computer Ihres lokalen Netzwerkes umsetzen. Normalerweise ist kaum oder gar keine Vorverarbeitung der Empfängeradressen erforderlich.

Regelsatz 4 ➤ Abschließende Nachverarbeitung der Adressen

Regelsatz 4 macht die von Regelsatz 3 vorgenommene Umsetzung teilweise wieder rückgängig. Die Konzentrierung der Adresse wird ebenfalls rückgängig gemacht, indem die spitzen Klammern um den Domainteil gelöscht werden. Beispielsweise könnte **amm<@rhino.uucp>** wieder in **rhino!amm** umgewandelt werden.

Informationsdienste

Je nach Systemkonfiguration verwendet **sendmail** verschiedene Quellen für Daten, die das gesamte Netzwerk betreffen ➤ darunter NetInfo, DNS, NIS und einige UNIX-Flachdateien. Die Verwendung hängt dabei jeweils von den benötigten Daten ab.

- Benutzernamen ➤ Wenn **sendmail** nach einem Benutzernamen sucht, wird zunächst NetInfo durchsucht. Wird der Name in NetInfo nicht gefunden und ist NIS aktiviert, wird die entsprechende UNIX-Datei untersucht. Anschließend wird die NIS-Tabelle geprüft.

- Hostnamen & Zusätzlich zu NetInfo und NIS wird auch DNS nach Hostnamen durchsucht. Sobald der Bestimmungsort einer Mitteilung ermittelt wurde, wird DNS danach durchsucht, ob ein *Mail-Vermittler* für den Bestimmungs-Computer vorhanden ist. Ein Mail-Vermittler ist ein Computer, der die gesamte Post bearbeitet, die an einen anderen Computer adressiert ist. Die Datensätze des Mail-Vermittlers in DNS werden *MX-Datensätze* genannt.

Wird ein MX-Datensatz für den Bestimmungs-Computer gefunden, wird dieser Computer bei der Postübertragung als Bestimmung verwendet (jedoch *nicht* in der Kopfzeile der Mitteilung). Die Daten im MX-Datensatz enthalten auch die Adresse des Mail-Vermittlers. Daher erfolgt kein späterer Bezug auf NetInfo oder NIS, um die Adresse des Mail-Vermittlers aufzulösen.

Die Reihenfolge, in der gesucht wird, ist NetInfo, DNS und NIS. Anschließend werden in DNS MX-Datensätze gesucht.

- Aliase & Aliase werden nur für Mitteilungen umgesetzt, die mit dem Mail-Programm **local** zugestellt werden. Um ein Alias aufzulösen, sucht **sendmail** die Adresse an drei Stellen, und zwar in dieser Reihenfolge: im NetInfo-Verzeichnis **/aliases**, in der NIS-Tabelle **mail.aliases** (wenn NIS läuft) und in der Datei, die mit der Option **A** der **sendmail**-Konfigurationsdatei angegeben wurde (im allgemeinen **/etc/sendmail/aliases**). Diese beiden letzten Möglichkeiten werden jedoch nur durchsucht, wenn die Adresse nicht bereits vorher gefunden wurde. Wird die Adresse überhaupt nicht gefunden, bleibt sie unverändert. Beachten Sie, daß dies unabhängig von der Mail-Programm-Flagge **I** gilt.

Nach der Alias-Auflösung sucht **sendmail** im Heimverzeichnis des Empfängers nach einer Datei namens **.forward**. Falls vorhanden, gibt diese Datei an, daß die Post weitergeleitet werden soll: an einen anderen Benutzer (nicht unbedingt lokal), an eine Datei oder an ein Programm.

In der folgenden Tabelle finden Sie Beispiele für alle möglichen Variationen der Datei **.forward**.

| Typ | Format | Beispiel |
|--------------|--------------------|--------------------------------|
| Benutzer | Mail-Adresse | amm@Tute.EDU |
| Literal user | Escaped address | \amm |
| Datei | Vollständiger Pfad | /Net/rhino/Users/amm/.maillog |
| Programm | <i>programm</i> | "/usr/local/bin/answermail -a" |

Wichtig: Falls eine Datei angegeben wird, muß sie bereits bestehen, damit Post hineingeschrieben werden kann & **sendmail** erstellt keine neuen Dateien.

Wird ein Befehl in einer Datei **.forward** angegeben, wird er im Spool-Verzeichnis der Mail-Warteschlange so ausgeführt, als hätte ihn der Mail-Empfänger eingegeben. Das Spool-Verzeichnis der Mail-Warteschlange wird durch die Option **Q** bestimmt (normalerweise **/usr/spool/mqueue**). Ist der Befehl ein Shell-Skript, wird **.profile** oder **.cshrc** des Benutzers ausgeführt. Wird der Befehl zusammen mit Argumenten eingegeben, muß die

gesamte Befehlszeile in Anführungszeichen stehen, wie im obigen Beispiel.

Möchten Sie die Post an mehrere Empfänger weiterleiten (z. B. an eine Datei und an ein Programm), geben Sie jeden Eintrag in eine separate Zeile der Datei **.forward** ein.

Warnung: Bei der Weiterleitung versucht **sendmail**, auf das Heimverzeichnis des Empfängers zuzugreifen. Befindet sich das Heimverzeichnis in einem importierten Dateisystem und ist der Datei-Server nicht verfügbar, muß der Prozeß **sendmail** eventuell warten, bis der Server wieder verfügbar ist.

Fehlersuche

Bei Problemen mit **sendmail** kann die Fehlersuche äußerst schwierig und komplex sein. Die Empfehlungen in diesem Abschnitt sollen Ihnen anfanglich helfen.

Die Optionen **mail -v** und **sendmail -v**

Zusammen mit der Option **-v** (verbose, ausführlich) eingegeben, zeigt der Berkeley-Befehl **mail** die **sendmail**-Dialoge oder -Aktionen an, die während der Postübertragung stattfinden. Nachfolgend finden Sie ein Beispiel dieses Befehls. Dabei erscheinen Benutzereingaben im Fettdruck und Kommentare in einem anderen Schrifttyp.

```
rhino [~]-175% mail -v rkabir
```

Post im ausführlichen Modus (verbose) an **rkabir** senden.

```
Subject: Testing
```

```
This is a test.
```

```
.
```

Eingabe einer Mitteilung, die mit einem Punkt auf einer separaten Zeile endet.

```
Cc:
```

Kopien an niemand.

```
rkabir... Connecting to mailhost via etherl...
```

Der lokale Computer versucht, Post an **rkabir** zu senden. Da dieser Computer ein Mail-Client ist, wird die Post an den Mail-Server (**mailhost**) geschickt. Dazu wird das Mail-Programm **etherl** verwendet.

```
Trying 192.42.172.66... connected.
```

192.42.172.66 ist die Internet-Adresse des Mail-Servers.

```
220 cockatoo.Rhino.COM Sendmail NX5.67c/NX3.0M ready at Wed, 15 Jul 92 09:14:18 -0700
```

Begrüßung durch den Mail-Server (**cockatoo**).

```
>>> HELO conure.Rhino.COM
```

```
250 cockatoo.Rhino.COM Hello conure.rhino.com, pleased to meet you
```

Die beiden Computer identifizieren sich gegenseitig.

```
>>> MAIL From:<amm>
```

```
250 <amm>... Sender ok
```

```

                Post vom Benutzer amm wird gesendet; die Absenderadresse wird akzeptiert.
>>> RCPT To:<rkabir>
250 <rkabir>... Recipient ok
                Post wird an rkabir gesendet; die Empf ngeradresse wird akzeptiert. Falls ein Problem mit dieser Adresse
                auftritt, wird eine Fehlermeldung ausgedruckt.

>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 Mail accepted
                Die Mail-Mitteilung wurde gesendet und empfangen.

>>> QUIT
221 cockatoo.Rhino.COM delivering mail
rkabir... Sent

```

Sie k nnen **sendmail** auch direkt mit der Option `^verbose^` aufrufen. Die Ausgabe ist mit der einer **Mail**-Sitzung im ausf hrlichen Modus identisch. Wenn Sie **sendmail** direkt ausf hren, m ssen Sie alle Kopfzeilen eingeben, die Ihr Postversandprogramm Ihnen zur Verf gung stellt (im allgemeinen **From:**, **To:**, **Cc:** und **Subject:**). Der Zustellungsversuch erfolgt an die Adressen, die in der Befehlszeile angegeben sind & unabh ngig von den Kopfzeilen.

So verwenden Sie `^sendmail^` im Adressentestmodus

Sie k nnen **sendmail** im *Adressentestmodus* ausf hren und so die Verarbeitungsergebnisse einer Adresse durch die verschiedenen Regels tze in der Konfigurationsdatei anzeigen. Im Adressentestmodus k nnen Sie die Regels tze angeben, die auf eine bestimmte Adresse angewendet werden sollen. Regels tze werden mit Zahlen angegeben und in Regelsatzfolgen durch Kommas voneinander getrennt. Weiterhin m ssen die Regels tze durch ein Leerzeichen von der Adresse getrennt werden. M chten Sie  berpr fen, wie **sendmail** eine bestimmte Adresse verarbeitet, geben Sie die Regelsatzfolge ein. Folgen Sie hierzu den Anweisungen weiter vorne in diesem Kapitel unter `^Flu^` und Anwendung der Regels tze.

Normalerweise wird die Adresse zun chst f r ein Mail-Programm aufgel st. Dazu lassen Sie die Umschlagempf ngeradresse (die Adresse des tats chlichen Empf ngers) durch Regelsatz 0 laufen. Die Umschlagadresse kann sich von der Adresse in der Kopfzeile unterscheiden. Erh lt z. B. ein Empf nger eine Blindkopie der Mitteilung, erscheint seine Adresse zwar auf dem Umschlag, aber nicht in der Kopfzeile. Normalerweise sind Umschlagadresse und Kopfzeilenadresse eines Empf ngers jedoch identisch.

Nachdem Sie f r eine Adresse Regelsatz 0 aufgerufen haben,  berpr fen Sie die Definition des Mail-Programms, das sich daraus ergibt. Bestimmen Sie damit, welche Regelsatzfolge jeweils f r die Absender- bzw. die Empf ngeradresse zu verwenden ist. Denken Sie daran, da  die Makros **S** und **R** einer Mail-Programm-Definition angeben, welche Regels tze auf die Absender- bzw. Empf ngeradresse anzuwenden sind.

Hinweis: Wenn Post zugestellt wird, führt **sendmail** normalerweise die Funktion ^aHinzufügen der Absender-Domain^o aus. Im Adressentestmodus führt **sendmail** diese Funktion jedoch nicht aus. Denken Sie daran, die Absender-Domain selbst allen Adressen hinzuzufügen, die zutreffen, wenn **sendmail** im Adressentestmodus läuft. Einzelheiten finden Sie weiter vorne in diesem Kapitel unter ^aRegelsatz D ^oHinzufügen der Absender-Domain^o.

Nachfolgend finden Sie eine Reihe kommentierter Sitzungen mit **sendmail** im Adressentestmodus, und zwar sowohl auf einem Mail-Client als auch auf einem Mail-Server. In den Beispielen erscheinen Benutzereingaben fettgedruckt und Kommentare in einem anderen Schrifttyp.

Adressentest auf einem Mail-Client

In der ersten Sitzung wird die **sendmail**-Verarbeitung mit Regelsätzen auf einem Mail-Client verfolgt. In diesem Beispiel wird vorausgesetzt, daß Sie Post von einem Computer namens **cockatoo** in der Domain **Rhino.COM** versenden. Unter den anderen Computern des lokalen Netzwerkes befinden sich der Mail-Server (**mailbox** mit dem Alias **mailhost**) und der Computer **conure**.

Denken Sie daran, daß Mail-Clients nichts über die Postzustellung wissen. Sie hängen vielmehr für Weiterleitungsdienste völlig vom Mail-Server ab.

Dies ist die Definition des Mail-Programms **etherl** aus der Datei **sendmail.sharedsubsidiary.cf**:

```
Metherl, P=[TCP], F=msDFuXn, S=12, R=22, A=TCP $h
```

Lokale Adresse

```
cockatoo [~]-329% /usr/lib/sendmail -bt
```

sendmail im Adressentestmodus ausführen.

```
ADDRESS TEST MODE
```

```
Enter <ruleset> <address>
```

```
[Note: No automatic ruleset 3 call]
```

Eine Begrüßung wird gedruckt. Im Gegensatz zu anderen **sendmail**-Programmen wird Regelsatz 3 nicht automatisch ausgeführt. Auf diese Weise können Sie einzelne Regelsätze ohne Beeinträchtigung durch Regelsatz 3 testen.

```
>3,0 amm
```

Regelsatz 3 und 0 auf **amm** anwenden, um die Adresse für ein Mail-Programm aufzulösen.

```
rewrite: ruleset 3 input: "amm"
```

```
rewrite: ruleset 3 returns: "amm"
```

```
rewrite: ruleset 0 input: "amm"
```

```
rewrite: ruleset 9 input: "amm"
```

```
rewrite: ruleset 9 returns: "amm"
```

```
rewrite: ruleset 0 returns: "$#" "etherl" "$@" "mailhost" " $:" "amm"
```

Regelsatz 3 wird aufgerufen, gemäß den Anweisungen gefolgt von Regelsatz 0. Regelsatz 0 ruft seinerseits Regelsatz 9 als Subroutine auf. Die Eingabe und die Ausgabe jedes Regelsatzes wird angezeigt. Regelsatz 0

gibt die Werte **etherl** (Mail-Programm), **mailhost** (Host, an den die Mitteilung gesendet werden soll) und **amm** (der Benutzer, der die Post empfangen soll) aus. Die Zeichenfolgen **\$#**, **\$@** und **\$:** besagen, daß die darauffolgenden Wörter sich auf das Mail-Programm, den Host oder den Benutzer beziehen.

> **3,1,12,4 tracy**

Die passende Regelsatzfolge auf die Absenderadresse (**tracy**) anwenden. Die Mail-Programm-Definition **etherl** gibt für den Absender Regelsatz 12 an (S=12).

```
rewrite: ruleset 3 input: "tracy"
rewrite: ruleset 3 returns: "tracy"
```

Ausführung von Regelsatz 3.

```
rewrite: ruleset 1 input: "tracy"
rewrite: ruleset 1 returns: "tracy"
```

Regelsatz 1 modifiziert die Adresse nicht.

```
rewrite: ruleset 12 input: "tracy"
rewrite: ruleset 12 returns: "tracy"
```

Der für das Mail-Programm spezifische Regelsatz 12 modifiziert die Adresse auch nicht.

```
rewrite: ruleset 4 input: "tracy"
rewrite: ruleset 9 input: "tracy"
rewrite: ruleset 9 returns: "tracy"
```

Regelsatz 4 ruft Regelsatz 9 auf.

```
rewrite: ruleset 4 returns: "tracy"
```

Die Adressen werden nicht verändert.

> **3,2,22,4 amm**

Die passende Regelsatzfolge auf die Empfängeradresse (**amm**) anwenden. Die Mail-Programm-Definition **etherl** gibt für den Empfänger Regelsatz 22 an (R=22).

```
rewrite: ruleset 3 input: "amm"
rewrite: ruleset 3 returns: "amm"
rewrite: ruleset 2 input: "amm"
rewrite: ruleset 2 returns: "amm"
rewrite: ruleset 22 input: "amm"
rewrite: ruleset 22 returns: "amm"
rewrite: ruleset 4 input: "amm"
rewrite: ruleset 9 input: "amm"
rewrite: ruleset 9 returns: "amm"
rewrite: ruleset 4 returns: "amm"
```

Wie bei der Absenderadresse waren auch hier keine ...nderungen notwendig.

Einfache entfernte Adresse

Als nächstes wird Post untersucht, die an eine entfernte Adresse gesendet wird (**amm** in der Anlage **Tute.EDU**). Auch dies ist ein Mail-Client, so daß die **sendmail**-Konfigurationsdatei die gleiche ist, einschließlich der Definition des Mail-Programms **etherl**.

> **3,0 amm@Tute.EDU**

Regelsätze 3 und 0 auf die Empfängeradresse anwenden, um sie für ein Mail-Programm aufzulösen.

```
rewrite: ruleset 3 input: "amm" "@" "tute" "." "EDU"  
rewrite: ruleset 6 input: "amm" "<" "@" "tute" "." "EDU" ">"
```

Regelsatz 3 führt Konzentrierung durch, bevor er Regelsatz 6 aufruft.

```
rewrite: ruleset 6 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 3 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 0 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 9 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 9 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 0 returns: "$#" "etherl" "$@" "mailhost" " $" "amm" "<" "@" "tute" "." "edu"  
">"
```

Die Adresse wird für das Mail-Programm **etherl** aufgelöst. Die Mitteilung würde an **mailhost** gesendet und an **amm@tute.edu** weitergeleitet.

> **3,1,12,4 tracy**

Die passende Regelsatzfolge auf die Absenderadresse anwenden. Erneut wird der für das Mail-Programm spezifische Absender-Regelsatz 12 definiert (R=12).

```
rewrite: ruleset 3 input: "tracy"  
rewrite: ruleset 3 returns: "tracy"  
rewrite: ruleset 1 input: "tracy"  
rewrite: ruleset 1 returns: "tracy"  
rewrite: ruleset 12 input: "tracy"  
rewrite: ruleset 12 returns: "tracy"  
rewrite: ruleset 4 input: "tracy"  
rewrite: ruleset 9 input: "tracy"  
rewrite: ruleset 9 returns: "tracy"  
rewrite: ruleset 4 returns: "tracy"
```

Keine ...nderungen - die Post wird an den Mail-Server weitergeleitet, wo eventuell ...nderungen vorgenommen werden.

> **3,2,22,4 amm@tute.edu**

Die Regelsatzfolge auf die Empfängeradresse anwenden. Regelsatz 22 wird in der Definition des Mail-Programms **etherl** angegeben (R=22).

```
rewrite: ruleset 3 input: "amm" "@" "tute" "." "edu"  
rewrite: ruleset 6 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 6 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 3 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 2 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 2 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 22 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 22 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 4 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 9 input: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 9 returns: "amm" "<" "@" "tute" "." "edu" ">"  
rewrite: ruleset 4 returns: "amm" "@" "tute" "." "edu"
```

Auch hier wurde die Adresse nicht verändert.

Beachten Sie, daß bei Post, die an den Mail-Server gesendet wird, die Adressen der Kopfzeile nie umgeschrieben werden.

UUCP-Adresse mit lokaler Verbindung

Untersuchen Sie nun die Umsetzung der Adresse einer Mitteilung, wenn diese über UUCP übertragen wird. Im nachfolgenden Beispiel wird vorausgesetzt, daß der Computer **ranger** ein UUCP-Partner von **cockatoo** ist. Nachstehend die Definition des Mail-Programms **uucp** aus der Datei **sendmail.sharedsubsidiary.cf**:

```
Muucp,      P=/usr/bin/uux, F=msDFMhuU, S=13, R=23, E=\n
            A=uux - -r $h!rmail ($u)
```

> **3,0 ranger!sandy**

Empfängeradresse für ein Mail-Programm auflösen.

```
rewrite: ruleset 3 input: "ranger" "!" "sandy"
rewrite: ruleset 6 input: "sandy" "<" "@" "ranger" "." "uucp" ">"
```

Kanonisation und Konzentrierung beendet.

```
rewrite: ruleset 6 returns: "sandy" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 3 returns: "sandy" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 0 input: "sandy" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 0 returns: "$#" "uucp" "$@" "ranger" " $" "sandy"
```

Diese Adresse wird für das Mail-Programm **uucp** aufgelöst und nicht an **mailhost** weitergeleitet. Es wird vorausgesetzt, daß auch Mail-Clients ihre eigenen UUCP-Partner haben können.

> **3,1,13,4 sandy**

Die passende Regelsatzfolge auf die Absenderadresse anwenden; das Mail-Programm **uucp** definiert Regelsatz 13 für Absenderadressen (S=13.)

```
rewrite: ruleset 3 input: "sandy"
rewrite: ruleset 3 returns: "sandy"
```

Kanonisation und Konzentrierung durch Regelsatz 3 (keine ...nderungen notwendig).

```
rewrite: ruleset 1 input: "sandy"
rewrite: ruleset 1 returns: "sandy"
```

Keine ...nderungen durch Regelsatz 1.

```
rewrite: ruleset 13 input: "sandy"
rewrite: ruleset 5 input: "sandy"
```

Regelsatz 13 ruft Regelsatz 5 auf, der die Umsetzung von UUCP-Adressen durchführt.

```
rewrite: ruleset 5 returns: "sandy"
rewrite: ruleset 13 returns: "cockatoo" "!" "sandy"
```

Regelsatz 13 hat der Adresse vorne **cockatoo!** hinzugefügt um anzuzeigen, daß die Absenderadresse in bezug auf den Empfangs-Computer **cockatoo!sandy** ist.

```
rewrite: ruleset 4 input: "cockatoo" "!" "sandy"
rewrite: ruleset 9 input: "cockatoo" "!" "sandy"
rewrite: ruleset 9 returns: "cockatoo" "!" "sandy"
```

```
rewrite: ruleset 4 returns: "cockatoo" "!" "sandy"
```

Da es sich von Anfang an um eine lokale Adresse handelte, war eine Konzentrierung nicht notwendig. Daher muß die Konzentrierung auch nicht durch Regelsatz 4 rückgängig gemacht werden. Außerdem hat die Adresse bereits ein kanonisches Format.

```
> 3,2,23,4 ranger!amm
```

Die Empfängeradresse umsetzen; die Definition des Mail-Programms **uucp** enthält R=23.

```
rewrite: ruleset 3 input: "ranger" "!" "amm"
rewrite: ruleset 6 input: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 6 returns: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 3 returns: "amm" "<" "@" "ranger" "." "uucp" ">"
```

Eine kanonische, konzentrierte Adresse wird zurückgegeben.

```
rewrite: ruleset 2 input: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 2 returns: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 23 input: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 5 input: "amm" "<" "@" "ranger" "." "uucp" ">"
rewrite: ruleset 5 returns: "ranger" "!" "amm"
rewrite: ruleset 23 returns: "ranger" "!" "amm"
rewrite: ruleset 4 input: "ranger" "!" "amm"
rewrite: ruleset 9 input: "ranger" "!" "amm"
rewrite: ruleset 9 returns: "ranger" "!" "amm"
rewrite: ruleset 4 returns: "ranger" "!" "amm"
```

An der Adresse sind keine weiteren ...nderungen notwendig.

Entfernte Adresse im lokalen Netzwerk

Im folgenden Beispiel wird untersucht, was passiert, wenn in einem lokalen Netzwerk die Post direkt an einen Computer gesendet wird und nicht an den Mail-Server. Es folgt die Definition des Mail-Programms **etherl** aus der Datei **sendmail.sharedsubsidiary.cf**:

```
Metherl, P=[TCP], F=msDFuXn, S=12, R=22, A=TCP $h
```

```
> 3,0 amm@cockatoo
```

```
rewrite: ruleset 3 input: "amm" "@" "cockatoo"
rewrite: ruleset 6 input: "amm" "<" "@" "cockatoo" ">"
```

Auf Hostteil konzentrieren.

```
rewrite: ruleset 6 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
```

Kanonisch umsetzen: die Adresse als lokal angeben.

```
rewrite: ruleset 3 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 0 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 30 input: "amm"
```

Regelsatz 30 ruft einfach Regelsatz 3 und danach Regelsatz 0 auf. In dieser lokalen Adresse sind überflüssige Leitwegdaten, dies wird von Regelsatz 0 bearbeitet. Da Sie sich in einem Netzwerk befinden und einen Mail-Server gemeinsam nutzen, wird vorausgesetzt, daß die gesamte Post an lokale Adressen an den Mail-Server gesendet wird. Andere Leitwegbestimmungen sind in einem lokalen Netzwerk überflüssig.

```
rewrite: ruleset 3 input: "amm"
```

Adressenauflösung neu beginnen, jetzt mit einer lokalen Adresse.

```
rewrite: ruleset 3 returns: "amm"
```

```
rewrite: ruleset 0 input: "amm"
```

```
rewrite: ruleset 9 input: "amm"
```

```
rewrite: ruleset 9 returns: "amm"
```

```
rewrite: ruleset 0 returns: "$#" "etherl" "$@" "mailhost" " $:" "amm"
```

```
rewrite: ruleset 30 returns: "$#" "etherl" "$@" "mailhost" " $:" "amm"
```

```
rewrite: ruleset 0 returns: "$#" "etherl" "$@" "mailhost" " $:" "amm"
```

Für das Mail-Programm **etherl** aufgelöst.

Wenn Sie die passenden Regelsätze auf die Kopfzeilenadressen anwenden, stellen Sie fest, daß keine Änderungen vorgenommen wurden. Wie bei den anderen Beispielen, in denen die Adresse aufgelöst wurde, um das Mail-Programm **etherl** zu verwenden.

Im folgenden Beispiel sehen Sie, was passiert, wenn die Post für einen Benutzer spezifisch an den Mail-Server adressiert wird:

```
> 3,0 amm@mailhost
```

```
rewrite: ruleset 3 input: "amm" "@" "mailhost"
```

```
rewrite: ruleset 6 input: "amm" "<" "@" "mailhost" ">"
```

```
rewrite: ruleset 6 returns: "amm" "<" "@" "LOCAL" ">"
```

Regelsatz 6 entdeckt, daß **mailhost** ein Alias für den lokalen Mail-Server ist, und zeigt an, daß es sich um eine lokale Adresse handelt.

```
rewrite: ruleset 3 returns: "amm" "<" "@" "LOCAL" ">"
```

```
rewrite: ruleset 0 input: "amm" "<" "@" "LOCAL" ">"
```

```
rewrite: ruleset 9 input: "amm" "<" "@" "rhino" "." "com" ">"
```

Regelsatz 0 wandelt **LOCAL** in **rhino.com** um - die Domain dieses Computers - und ruft dann Regelsatz 9 auf.

```
rewrite: ruleset 9 returns: "amm" "<" "@" "rhino" "." "com" ">"
```

```
rewrite: ruleset 0 returns: "$#" "etherl" "$@" "mailhost" " $:" "amm" "<" "@" "rhino" "." "com" ">"
```

Die Post geht erwartungsgemäß an den Mail-Server. Die Umschlagadresse lautet jedoch **amm@rhino.com**. Sie können voraussetzen, daß der Mail-Server eine solche Adresse als lokale Adresse erkennt.

An den Kopfzeilenadressen erkennen Sie, daß die Absenderadresse unverändert ist, die Empfängeradresse jedoch modifiziert wurde.

```
> 3,2,22,4 amm@mailhost
```

Regelsatzfolge auf Empfängeradresse anwenden.

```
rewrite: ruleset 3 input: "amm" "@" "mailhost"
```

```
rewrite: ruleset 6 input: "amm" "<" "@" "mailhost" ">"
```

```
rewrite: ruleset 6 returns: "amm" "<" "@" "LOCAL" ">"
```

```
rewrite: ruleset 3 returns: "amm" "<" "@" "LOCAL" ">"
```

Die Adresse ist jetzt konzentriert und kanonisch umgesetzt.

```
rewrite: ruleset 2 input: "amm" "<" "@" "LOCAL" ">"
```

```
rewrite: ruleset 2 returns: "amm" "<" "@" "LOCAL" ">"
rewrite: ruleset 22 input: "amm" "<" "@" "LOCAL" ">"
rewrite: ruleset 22 returns: "amm"
```

Der für das Mail-Programm spezifische Regelsatz 22 setzt in diesem Fall **amm@LOCAL** in **amm** um.
Dadurch erscheint die Kopfzeilenadresse wie eine lokale Adresse.

```
rewrite: ruleset 4 input: "amm"
rewrite: ruleset 9 input: "amm"
rewrite: ruleset 9 returns: "amm"
rewrite: ruleset 4 returns: "amm"
```

Adressentest auf einem Mail-Server

Sie haben gesehen, was mit den Mitteilungen und Adressen auf einem Mail-Client passiert. Nachfolgend können Sie untersuchen, was mit Ähnlichen Adressen auf einem Mail-Server geschieht. Im Beispiel wird davon ausgegangen, daß der Server an das Internet angeschlossen ist und eine Doppelfunktion hat: Er ist sowohl der lokale Mail-Server als auch der Internet-Mail-Gateway. Der Name des Mail-Clients ist wieder **cockatoo**, der Name des Mail-Servers ist **mailbox**.

Nachfolgend die Definition des Mail-Programms **local** aus der Datei **sendmail.mailhost.cf**:

```
Mlocal,      P=/bin/mail, F=rlsDFMnP, S=10, R=20, A=mail -d $u
```

Lokale Adresse

```
mailbox [~]-20% /usr/lib/sendmail -bt
ADDRESS TEST MODE
Enter <ruleset> <address>
[Note: No automatic ruleset 3 call]
> 3,0 amm
rewrite: ruleset 3 input: "amm"
rewrite: ruleset 3 returns: "amm"
rewrite: ruleset 0 input: "amm"
rewrite: ruleset 9 input: "amm"
rewrite: ruleset 9 returns: "amm"
rewrite: ruleset 0 returns: "$#" "local" " $:" "amm"
```

Die Adresse wird für das Mail-Programm **local** aufgelöst.

```
> 3,1,10,4 tracy
```

Das Mail-Programm **local** definiert seinen Absender-Regelsatz (S=) als 10.

```
rewrite: ruleset 3 input: "tracy"
rewrite: ruleset 3 returns: "tracy"
rewrite: ruleset 1 input: "tracy"
rewrite: ruleset 1 returns: "tracy"
```

```

rewrite: ruleset 10 input: "tracy"
rewrite: ruleset 10 returns: "tracy"
rewrite: ruleset 4 input: "tracy"
rewrite: ruleset 9 input: "tracy"
rewrite: ruleset 9 returns: "tracy"
rewrite: ruleset 4 returns: "tracy"

```

> **3,2,20,4 amm**

Das Mail-Programm **local** definiert seinen Empfänger-Regelsatz (R=) als 20.

```

rewrite: ruleset 3 input: "amm"
rewrite: ruleset 3 returns: "amm"
rewrite: ruleset 2 input: "amm"
rewrite: ruleset 2 returns: "amm"
rewrite: ruleset 20 input: "amm"
rewrite: ruleset 20 returns: "amm"
rewrite: ruleset 4 input: "amm"
rewrite: ruleset 9 input: "amm"
rewrite: ruleset 9 returns: "amm"
rewrite: ruleset 4 returns: "amm"

```

Erwartungsgemäß wurden weder die Absenderadresse noch die Empfängeradresse verändert.

Einfache entfernte Adresse

Im folgenden Beispiel wird Post an einen entfernten Computer geschickt. Die Absenderadresse ist **amm**, diesmal in **Tute.EDU**. Nachfolgend die Definition des Mail-Programms **ddn** aus der Datei **sendmail.mailhost.cf**:

```
Mddn,      P=[TCP], F=msDFMuCX, S=22, R=22, A=TCP $h, E=\r\n
```

> **3,0 amm@Tute.EDU**

```

rewrite: ruleset 3 input: "amm" "@" "Tute" "." "EDU"
rewrite: ruleset 6 input: "amm" "<" "@" "Tute" "." "EDU" ">"
rewrite: ruleset 6 returns: "amm" "<" "@" "tute" "." "edu" ">"
rewrite: ruleset 3 returns: "amm" "<" "@" "tute" "." "edu" ">"
rewrite: ruleset 0 input: "amm" "<" "@" "tute" "." "edu" ">"
rewrite: ruleset 9 input: "amm" "<" "@" "tute" "." "edu" ">"
rewrite: ruleset 9 returns: "amm" "<" "@" "tute" "." "edu" ">"
rewrite: ruleset 0 returns: "$#" "ddn" "$@" "tute" "." "edu" "$:" "amm" "<" "@" "tute" "."
"edu" ">"

```

Die Adresse wird für das Mail-Programm **ddn** aufgelöst.

> **3,1,22,4 tracy**

Das Mail-Programm **ddn** definiert seinen Absender-Regelsatz als 22.

```

rewrite: ruleset 3 input: "tracy"
rewrite: ruleset 3 returns: "tracy"
rewrite: ruleset 1 input: "tracy"
rewrite: ruleset 1 returns: "tracy"

```

Würde Regelsatz 1 hier eine Umsetzung in eine standardmäßige (kanonische) Mail-Adresse wie etwa **Tracy_Roykirk** vornehmen, wäre die Ausgabe hier statt dessen **tracy_roykirk**.

```
rewrite: ruleset 22 input: "tracy"
```

```
rewrite: ruleset 22 returns: "tracy" "<" "@" "mailbox" "." "rhino" "." "com" ">"
```

Regelsatz 22 fügt den voll qualifizierten Hostnamen hinzu. Er führt außerdem die gerade erwähnte standardmäßige (kanonische) Umsetzung durch. (Weitere Einzelheiten enthält **sendmail.mailhost.cf**). In diesem Fall wird davon ausgegangen, daß eine standardmäßige Umsetzung nur bei Mitteilungen vorgenommen wird, die an entfernte Computer gesendet werden. Sie brauchen den Hostnamen des Mail-Servers nicht unbedingt mit anzugeben und können nur **rhino.com** als Rückanlage verwenden. (Dabei müssen Sie sich jedoch vergewissern, daß **rhino.com** vom externen Netzwerk als **mailbox.rhino.com** verstanden wird.)

```
rewrite: ruleset 4 input: "tracy" "<" "@" "mailbox" "." "rhino" "." "com" ">"
```

```
rewrite: ruleset 9 input: "tracy" "<" "@" "mailbox" "." "rhino" "." "com" ">"
```

```
rewrite: ruleset 9 returns: "tracy" "<" "@" "mailbox" "." "rhino" "." "com" ">"
```

```
rewrite: ruleset 4 returns: "tracy" "@" "mailbox" "." "rhino" "." "com"
```

> **3,2,22,4 amm@Tute.EDU**

Das Mail-Programm **ddn** definiert seinen Empfänger-Regelsatz als 22, genau wie den Absender-Regelsatz.

```
rewrite: ruleset 3 input: "amm" "@" "Tute" "." "EDU"
```

```
rewrite: ruleset 6 input: "amm" "<" "@" "Tute" "." "EDU" ">"
```

```
rewrite: ruleset 6 returns: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 3 returns: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 2 input: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 2 returns: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 22 input: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 22 returns: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 4 input: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 9 input: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 9 returns: "amm" "<" "@" "tute" "." "edu" ">"
```

```
rewrite: ruleset 4 returns: "amm" "@" "tute" "." "edu"
```

Die Empfänger-Adresse wurde nicht verändert.

Entfernte Adresse im lokalen Netzwerk

Auf einem Mail-Server ist die Sache nicht ganz so einfach, auch mit einfachen entfernten Adressen. In folgenden Beispielen wird die Post ausdrücklich an den Mail-Server gesendet (vom Mail-Server) und danach ausdrücklich an einen anderen lokalen Host. Es folgt die Definition des Mail-Programms **etherl** aus der Datei **sendmail.mailhost.cf**:

```
Mether,      P=[TCP], F=msDFMuCX, S=11, R=21, A=TCP $h
```

> **3,0 amm@mailhost**

Den Host-Alias **mailhost** und nicht ausdrücklich den Hostnamen des Mail-Servers verwenden. Beide haben die gleiche Wirkung.

```

rewrite: ruleset 3 input: "amm" "@" "mailhost"
rewrite: ruleset 6 input: "amm" "<" "@" "mailhost" ">"
rewrite: ruleset 6 returns: "amm" "<" "@" "LOCAL" ">"
rewrite: ruleset 3 returns: "amm" "<" "@" "LOCAL" ">"
rewrite: ruleset 0 input: "amm" "<" "@" "LOCAL" ">"
rewrite: ruleset 30 input: "amm"
rewrite: ruleset 3 input: "amm"
rewrite: ruleset 3 returns: "amm"
rewrite: ruleset 0 input: "amm"
rewrite: ruleset 9 input: "amm"
rewrite: ruleset 9 returns: "amm"
rewrite: ruleset 0 returns: "$#" "local" " $:" "amm"
rewrite: ruleset 30 returns: "$#" "local" " $:" "amm"
rewrite: ruleset 0 returns: "$#" "local" " $:" "amm"

```

sendmail erkennt, daß **mailhost** ein Name des aktuellen Computers ist. Die Spezifikation **@mailhost** wird gelöscht, und Regelsätze 3 und 0 werden nochmals auf die Adresse angewendet. Danach wird die Adresse für das lokale Mail-Programm aufgelöst.

> **3,0 amm@cockatoo**

Empfängeradresse ausdrücklich an einen anderen Host im Netzwerk.

```

rewrite: ruleset 3 input: "amm" "@" "cockatoo"
rewrite: ruleset 6 input: "amm" "<" "@" "cockatoo" ">"
rewrite: ruleset 6 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 3 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 0 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 0 returns: "$#" "ether" "$@" "cockatoo" " $:" "amm" "<" "@" "cockatoo" ">"

```

Die Post wird mit dem Mail-Programm **ether** an **cockatoo** gesendet.

> **3,1,11,4 tracy**

Das Mail-Programm **ether** definiert den Absender-Regelsatz als 11.

```

rewrite: ruleset 3 input: "tracy"
rewrite: ruleset 3 returns: "tracy"
rewrite: ruleset 1 input: "tracy"
rewrite: ruleset 1 returns: "tracy"
rewrite: ruleset 11 input: "tracy"
rewrite: ruleset 11 returns: "tracy" "<" "@" "mailbox" ">"

```

Regelsatz 11 fügt den Hostnamen dieses Computers hinzu.

```

rewrite: ruleset 4 input: "tracy" "<" "@" "femail" ">"
rewrite: ruleset 9 input: "tracy" "<" "@" "femail" ">"
rewrite: ruleset 9 returns: "tracy" "<" "@" "femail" ">"
rewrite: ruleset 4 returns: "tracy" "@" "femail"

```

Da Sie ausdrücklich einen Hostnamen in der Adresse der Mitteilung verwendet haben, sollte die Rückadresse (die des Absenders) den ausdrücklichen Hostnamen des Absender-Computers enthalten.

> **3,2,21,4 amm@cockatoo**

Regelsatz 21 wird in der Definition des Mail-Programms **ether** als Regelsatz der Empfängeradresse angegeben.

```

rewrite: ruleset 3 input: "amm" "@" "cockatoo"
rewrite: ruleset 6 input: "amm" "<" "@" "cockatoo" ">"
rewrite: ruleset 6 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
        Bekannter Computer im lokalen Netzwerk erkannt, .LOCAL angefügt.
rewrite: ruleset 3 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 2 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 2 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 21 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 21 returns: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 4 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 9 input: "amm" "<" "@" "cockatoo" "." "LOCAL" ">"
rewrite: ruleset 9 returns: "amm" "<" "@" "cockatoo" "." "rhino" "." "com" ">"
        Regelsatz 9 wandelt die Pseudo-Domain LOCAL in die echte Domain um.
rewrite: ruleset 4 returns: "amm" "@" "cockatoo" "." "rhino" "." "com"

```

Konfigurationsdateien auf NeXT-Computern

NeXT-Computer werden mit drei verschiedenen **sendmail**-Konfigurationsdateien geliefert. Jede dient spezifischen Verwendungszwecken:

- **sendmail.subsidiary.cf** Ð Wird bei Computern verwendet, die Post lokal in einem Spool-Speicher aufbewahren, aber keine kompletten Mail-Leitwegdaten haben. Der Computer ist weder Mail-Client noch Mail-Server. Dies ist die voreingestellte Konfigurationsdatei.
- **sendmail.mailhost.cf** Ð Wird auf Mail-Servern verwendet.
- **sendmail.sharedsubsidiary.cf** Ð Diese Datei wird auf Client-Computern verwendet, die das Spool-Verzeichnis von einem Server importieren.

Makros

In der folgenden Tabelle wird beschrieben, wie Makros definiert und in der standardmÜûigen **sendmail**-Konfigurationsdatei auf NeXT-Computern verwendet werden.

| Makro | Verwendung | Definition | Verwendung bei |
|-------|----------------------|-------------------------|--|
| M | Relais-Mail-Programm | uucp ether etherl | mailhost subsidiary sharedsubsidiary |

| | | | |
|---|---------------------------------|--|--|
| R | Relais | hostmail-relay mailhost mailhost | mailhost subsidiary sharedsubsidiary |
| V | Version der Konfigurationsdatei | NX3.0M NX3.0S NX3.0X | mailhost subsidiary sharedsubsidiary |
| Z | Kanonische Adreûtabelle | mail.byaddr | mailhost |
| e | SMTP-Begröûung | \$j Sendmail \$v/\$V ready at \$b | alle |
| j | Hostname | \$?m \$w.\$m \$ \$w \$. | alle |
| l | Format für Kopfzeile ªFromº | From \$g \$d | alle |
| n | Medium des Mail-Programms | Mailer-Agent | alle |
| o | Operatoren in Adressen | .:%@!^=/[] | alle |
| q | Absenderadresse | Dq\$?x\$x \$.<\$g> | alle |

Einige dieser Makros werden in allen Konfigurationsdateien in der gleichen Weise definiert; andere werden in jeder Datei unterschiedlich definiert und wieder andere werden überhaupt nur in bestimmten Dateien definiert.

Mit dem Makro **Z** wird der Name einer NIS-Tabelle für ªumgekehrte Aliasumsetzungº angezeigt. Angenommen, ein Benutzer mit dem ªloginº-Namen **rkabir** sendet Post. In diesem Fall enthÛlt die NIS-Tabelle, auf die das Makro **Z** verweist, einen Eintrag, mit dem **rkabir** in einen anderen Namen umgesetzt wird Ð wie etwa **Randy_Kabir**. Wenn NIS nicht lÛuft oder wenn die Bezugstabelle nicht definiert wurde, erfolgt keine Umsetzung (und kein Fehler).

Mit dem Makro **o** werden die Operatoren definiert, die in Adressen verwendet werden. Die Operatoren trennen Zeichenfolgen voneinander.

Tip: Falls Sie Ihre Konfigurationsdatei Ûndern, sollten Sie gleichzeitig eine kleine ...nderung an der Definition des Makros **V** vornehmen und so angeben, daß die Datei verÛndert wurde. Auf diese Weise können die ...nderungen von Ihnen verfolgt und von anderen Benutzern einfacher erkannt werden.

In der folgenden Tabelle werden Musterwerte für solche Makros beschrieben, die durch andere Makros definiert werden.

| Makro | Musterwert |
|--------------|--|
| e | rhino.Tute.EDU Sendmail NX5.67c/NX3.0X ready atWed, 15 Jul 92 14:03:06 -0700 |
| j | rhino.Tute.EDU |
| l | From amm Fri Jan 3 13:54:40 1992 |
| q | amm |

Klassen

In der folgenden Tabelle werden die Klassen beschrieben, die in den Konfigurationsdateien definiert werden. Beachten Sie, daß eine Klasse zwar den gleichen Namen wie ein Makro haben kann, Makros und Klassen aber völlig unterschiedliche Werte haben und auch auf verschiedene Weise verwendet werden. Eine Konfigurationsdatei wird jedoch wesentlich verständlicher, wenn Makros und Klassen mit dem gleichen Namen auch in ähnlicher Weise verwendet werden.

| Klasse | Verwendung | Definition | Verwendung bei |
|---------------|-------------------|------------------------------------|--|
| R | Relais-Host | mail-relay mailhost mailhost | mailhost subsidiary sharedsubsidiary |
| V | UUCP-Partner | Ausgabe von /usr/bin/uuname | alle |

Beachten Sie, daß die Klasse **R** mit dem gleichen Wert wie das Makro **R** definiert wurde. Beide speichern den Namen (oder die Namen, im Fall der Klasse) des Mail-Relais-Hosts. Die Klasse **R** definiert alle Namen des Relais-Hosts, einschließlich der Aliase. Dagegen definiert das Makro **R** den bevorzugten Namen des Relais-Hosts. Beide werden benötigt, da sie verschieden verwendet werden. In den Klassen werden gleichwertige Elemente aufgelistet, während die Makros Variablen sind, die für einfache Substitutionen verwendet werden.

Mit der Klasse **R** können Sie feststellen, ob ein Host mit dem Relais-Host ^aäquivalent^o ist. Dazu verwenden Sie das Metasymbol **\$=** in der linken Seite einer Regel. Ist der Host äquivalent, wollen Sie die Post vielleicht anders verarbeiten. (Beachten Sie, daß die Klasse **R** zwar definiert ist, aber nicht in den voreingestellten Konfigurationsdateien verwendet wird.)

Adressenauflösung

Mail-Adressen können rein lokale Adressen sein (z. B. **benutzer**). Sie können aber auch ausdrücklich einen anderen lokalen Host (z. B. **benutzer@host**) angeben oder einen entfernten Host betreffen (etwa **benutzer@host.domain**). In der folgenden Tabelle wird angezeigt, wie die verschiedenen Adressen für jede der drei Konfigurationsdateien aufgelöst werden.

| Adresse | mailhost | subsidiary | sharedsubsidiary |
|------------------|----------|-------------------|--------------------|
| user | local | local | etherl an mailhost |
| user@host | ether | ether | etherl an mailhost |
| user@host.domain | ddn | ether an mailhost | ether an mailhost |

Sehen Sie sich die Zeile für **benutzer@host.domain** an. Wird diese Adresse auf dem Mail-Server aufgelöst (mit **sendmail.mailhost.cf**), wird das Mail-Programm **ddn** verwendet. Wird die Adresse auf einem Computer aufgelöst, der **sendmail.subsidiary.cf** verwendet, wird sie mit dem Mail-Programm **ether** an den Computer **mailhost** weitergeleitet. Sie wird auch mit dem Mail-Programm **ether** an **mailhost** gesendet, wenn sie auf einem Mail-Client mit **sendmail.sharedsubsidiary.cf** aufgelöst wird.

Regelsätze

In den Konfigurationsdateien eines NeXT-Computers werden die verschiedenen Regelsätze einheitlich verwendet. Der Regelsatz 5 in **sendmail.mailhost.cf** dient dem gleichen Zweck wie der Regelsatz 5 in **sendmail.sharedsubsidiary.cf**. In den folgenden drei Tabellen wird die Verwendung der verschiedenen Regelsätze zusammengefaßt. Die Regelsätze sind in drei Gruppen unterteilt: standardmäßige Regelsätze, nicht-standardmäßige Regelsätze, die auch keine für Mail-Programme spezifischen Umschreiberegelsätze sind, und schließlich die für Mail-Programme spezifischen Umschreiberegelsätze. Die folgende Tabelle beschreibt die standardmäßigen Regelsätze.

| Regelsatz | Zweck |
|-----------|--|
| 0 | Mail-Programm-Auflösung |
| 1 | Standardmäßige Vorverarbeitung der Absender-Kopfzeilenadresse |
| 2 | Standardmäßige Vorverarbeitung der Empfänger-Kopfzeilenadresse |
| 3 | Kanonisation und Konzentrierung |
| 4 | Standardmäßige Nachverarbeitung der Kopfzeilenadresse |

Die folgende Tabelle zeigt nicht-standardmäßige Regelsätze, die von keinem Mail-Programm verwendet werden, um Adressen umzuschreiben. Diese Regelsätze werden normalerweise von anderen Regelsätzen für häufig durchgeführte Operationen aufgerufen.

| Regelsatz | Zweck |
|-----------|--|
| 5 | Konvertiert benutzer@host.uucp in host!benutzer |
| 6 | Integration verschiedener lokaler Konventionen |
| 9 | AdressensÜberung, bevor die Adressen an ein Mail-Programm weitergegeben werden |
| 30 | Föhrt Regelsatz 3 gefolgt von Regelsatz 0 aus |

In der nÜchsten Tabelle werden die für ein Mail-Programm spezifischen UmschreibregelsÜtze zusammengefaût. Regelsatz 12 wird nur von Mail-Clients und Regelsatz 22 nur von Mail-Servern und Mail-Clients verwendet (und nicht in **sendmail.subsidiary.cf**).

| Regelsatz | Zweck |
|-----------|---|
| 10 | Umschreibregelsatz für Absenderadressen der Mail-Programme local und prog |
| 20 | Umschreibregelsatz für EmpfÜngeradressen der Mail-Programme local und prog |
| 11 | Umschreibregelsatz für Absenderadressen des Mail-Programms ether |
| 21 | Umschreibregelsatz für EmpfÜngeradressen des Mail-Programms ether |
| 12 | Umschreibregelsatz für Absenderadressen des Mail-Programms etherl |
| 22 | Umschreibregelsatz für EmpfÜngeradressen des Mail-Programms etherl (Mail-Client) Umschreibregelsatz für Absender- und EmpfÜngeradressen des Mail-Programms ddn (Mail-Server) |
| 13 | Umschreibregelsatz für Absenderadressen des Mail-Programms uucp |
| 23 | Umschreibregelsatz für EmpfÜngeradressen des Mail-Programms uucp |

Branchenkniffe

Eine derartig massive Herausforderung wie die Unterstützung von **sendmail** ist nichts für zaghafte Gemöter. In diesem Abschnitt finden Sie einige grundlegende Empfehlungen für eine erfolgreiche Anwendung von **sendmail**.

Allgemeine RatschlÜge

- Wenden Sie sich an einen erfahrenen Berater, der Ihnen dabei hilft, sich mit **sendmail** vertraut zu machen. Ihre Kontaktperson sollte sich gut mit **sendmail** auskennen und gerne auf alle Ihre Fragen antworten.
- Für diejenigen, die mehr in die Tiefe gehen möchten, ist eventuell Quellcode über anonymes FTP verfügbar & beispielsweise von **uunet.uu.net** (in **/mail/sendmail**).

Warnung: Allgemeine **sendmail**-Quellcodes, die in den Archiven verfügbar sind, können in dieser Form höchstwahrscheinlich nicht auf NeXT-Computern verwendet werden. Verschiedene NeXT-spezifische Modifikationen wurden vorgenommen & insbesondere zur Unterstützung von NetInfo.

- Vielleicht finden Sie auch manchen hilfreichen Beitrag in den Diskussionen der UseNet-Nachrichtengruppe **comp.mail.sendmail**. Außerdem können andere Leser vielleicht auf Ihre Fragen antworten und Ihnen helfen.

Kniffe beim Betrieb

Es gibt drei wichtige Kniffe, die Ihnen in bestimmten Betriebssituationen helfen können.

Post für Aliase einschließlich des Absenders

Wenn ein Benutzer Post an einen Mail-Alias schickt, in dem nach der Auflösung der Account-Name des Absenders enthalten ist, erhält der Absender normalerweise keine Kopie der Mitteilung. Wenn in der **sendmail**-Konfigurationsdatei jedoch die Option **m** enthalten ist, erhält der Absender in diesem Fall eine Kopie seiner Mitteilung. Beachten Sie, daß die Einstellungen einer **sendmail**-Konfigurationsdatei auf dem Mail-Client nicht ausschlaggebend sind. Ausschlaggebend sind die Einstellungen in der Konfigurationsdatei des Mail-Servers. Sie bestimmen, ob der Absender eine Kopie der Mail-Mitteilung erhält.

Einfügen von ^arealname^o in die Adresse ^aFrom:^o

Mit der Definition des Makros **q** in der Konfigurationsdatei des Absenders wird festgelegt, welches Format die Adresse **From:** in der Kopfzeile hat. Die Definition in **sendmail.sharedsubsidiary.cf** schließt das Feld **realname** nicht ein. Wenn Sie möchten, daß dieses Feld in die Adresse **From:** mit eingeschlossen wird, müssen Sie die Definition wie folgt ändern:

```
Dq$?x$x$.<$g>
```

Mit dieser Definition hat die Rückadresse etwa folgendes Format:

```
troykirk (Tracy Q. Roykirk)
```

Post für unbekannte Domains

Wie weiter vorne erläutert, definieren das Makro **M**, die Klasse **R** und das Makro **R** das Relais-Mail-Programm und den Relais-Computer. In der Konfigurationsdatei **sendmail.mailhost.cf** werden sie als **uucp**, **mail-relay** und **mail-relay** definiert. In der Voreinstellung werden diese Werte nicht verwendet. Statt dessen wird Post für unbekannte Domains an das Mail-Programm **ddn** gesendet, also direkt an die unbekannte Domain. Wenn Ihr Mail-Server nicht mit dieser Domain verbunden ist, wird die Mitteilung als unzustellbar zurückgesendet.

Soll die Post für unbekannte Domains anders behandelt werden, müssen Sie das Makro **M**, die Klasse **R** und das Makro **R** in **sendmail.mailhost.cf** mit den entsprechenden Werten definieren. Anschließend modifizieren Sie in derselben Datei Regelsatz 0. Die Zeilen, die geändert werden müssen, beginnen bei Version NX3.0M mit Zeile 338:

```
# If you want to pass all other explicit domain names up the ladder
# to our forwarder then uncomment the following line.#
#R$*<@$*.$+>$*    $#$M $@$R $:$1<@$2.$3>$4    user@any.domain
# and comment out this one.#
R$*<@$+.$->$*      $#ddn $@ $2.$3 $:$1<@$2.$3>$4      user@any.domain
```

Nehmen Sie die ...änderungen vor, die in den Kommentaren beschrieben werden. Das Ergebnis sieht so aus:

```
# If you want to pass all other explicit domain names up the ladder# # to our forwarder then
uncomment the following line.#
R$*<@$*.$+>$*      $#$M      $@$R $:$1<@$2.$3>$4      user@any.domain
# and comment out this one.#
#R$*<@$+.$->$*      $#ddn $@ $2.$3 $:$1<@$2.$3>$4      user@any.domain
```

Kniffe bei der Fehlersuche

Wenn Sie bei einem Konfigurationsproblem mit **sendmail** einen Debuglauf durchführen, möchten Sie eventuell die Werte verschiedener Makros untersuchen. So können Sie die Art und Weise der Definition genau bestimmen. Sie können Ihren Konfigurationsdateien zu diesem Zweck neue Einträge hinzufügen.

So fügen Sie einen Regelsatz hinzu, um Makros zu untersuchen

In der folgenden Definition des Regelsatzes 16 wird das Mail-Programm **error** verwendet, um die Werte der Makros **j** und **m** anzuzeigen.

```
S16
R$*    $#error $:j=$j, m=$m
```

Führen Sie **sendmail** im Adressentestmodus aus und rufen Sie Regelsatz 16 auf:

```
> 16 test
rewrite: ruleset 16    input: "test"
rewrite: ruleset 16 returns: "$#" "error" " $" "j" "=" "cockatoo" "." "rhino" "." "com" ", "
"m" "=" "rhino" "." "com"
```

Normalerweise wird das Mail-Programm **error** nur im Regelsatz 0 aufgerufen. In diesem Fall bietet es jedoch einen einfachen Mechanismus, mit dem Meldungen gedruckt werden. Das Makro **j** hat den Wert **cockatoo.rhino.com**, und das Makro **m** hat als Wert **rhino.com**.

So fügen Sie Kopfzeilen hinzu, um Makros zu untersuchen

Einige Makros enthalten keine sinnvollen Werte, wenn die Mitteilung noch nicht verarbeitet wurde. Dazu zählen etwa **u** (die Empfängeradresse) oder **q** (das Format einer Absenderadresse). Diese Makros können Sie mit keinem Regelsatz untersuchen. Statt dessen können Sie die Kopfzeilen modifizieren, um diese Werte anzuzeigen.

Fügen Sie den Kopfzeilendefinitionen die folgenden Zeilen hinzu:

```
HX-Test1: j=$j, m=$m
HX-Test2: u=$u, q=$q
```

Wenn vor dem Namen einer nicht-standardmäßigen Kopfzeile **X-** steht, entspricht dies den allgemeinen Konventionen über Mail-Kopfzeilen. Außerdem entspricht Ihre Mitteilung auf diese Weise allen anderen wichtigen Normen.

Senden Sie sich selbst eine kurze Test-Mitteilung. Es erscheinen die beiden folgenden zusätzlichen Kopfzeilen:

```
X-Test1: j= cockatoo.rhino.com, m=rhino.com
X-Test2: u=tr, q=<tracy>
```

Die Mitteilung wurde vom Benutzer **tracy** an den Alias **tr** gesendet.

So zeigen Sie alle Kopfzeilen an

Die Anwendung ^aNeXTmail^o zeigt in einer Mitteilung nur einige der Kopfzeilen an. Wenn die gewünschten

Kopfzeilen im Mitteilungsfenster nicht angezeigt werden, gehen Sie folgendermaßen vor, um alle Kopfzeilen der Mitteilung aufzulisten:

1. Klicken Sie im Briefkasten-Fenster auf den Kopf der Mitteilung.
2. Kopieren Sie die Mitteilung mit `^Command-c`, *ohne* jedoch in den Mitteilungsbereich zu klicken.
3. Fügen Sie die Mitteilung beispielsweise in ein neues Fenster der Anwendung `^Edit` ein.

Falls die Mitteilung ein reiner ASCII-Text ist, erscheint sie vollständig im Fenster der Anwendung `^Edit`. Enthält die Mitteilung dagegen Text im RTF-Format, eine Anlage, eine Sprachaufzeichnung oder ein Bild, sehen Sie nur die Kopfzeilen.

Es folgt ein Beispiel mit allen Kopfzeilen einer Mitteilung, die von Tracy Roykirk vom Computer **cockatoo** aus gesendet wurde. Die Mitteilung wurde an **mailbox** weitergeleitet und von dort dem Benutzer **amm** zugestellt.

```
From Tracy_Roykirk Wed Jul 15 13:57:41 1992
Next-Reference: Testing.attach, 1/1
Return-Path: <Tracy_Roykirk>
Received: from cockatoo.rhino.com (cockatoo) by mailbox.Rhino.COM (NX5.67c/NX3.0M)
        id AA00320; Wed, 15 Jul 92 13:57:35 -0700
From: Tracy_Roykirk (Tracy Q. Roykirk - SysAdmin)
Message-Id: <9112270031.AA00320@mailbox.Rhino.COM>
Received: by cockatoo.rhino.com (NX5.67c/NX3.0X)
        id AA00937; Wed, 15 Jul 92 13:57:30 -0700
Date: Wed, 15 Jul 92 13:57:35 -0700
Received: by NeXT Mailer (1.85)
Received: by NeXT Mailer (1.85)
To: amm
Subject: Testing
```