Mail

A JvNCnet Perspective

Vikas Aggarwal

January 12, 1991
Last update: Dec 13, 1991

Contents

Abstract

Discusses mail policies and alternatives. Different mailers, configuring sendmail for different situations. Excerpts from the tcp-ip mailing list discussing the strategies that different sites use.

1 Overview

Mail has always been one of the main uses of computer networks. The entire system has gotten more and more complex as the number and type of systems has become more diverse. Interaction and interoperatibilty between different mail systems (MCI mail, Usenet, Bitnet, Internet) has become of prime importance, and there are a number of mail alternatives available to a site - not only are there a number of mail software available, but there are a variety of strategies and policies that a site must decide on.

This document is centered about its topic – mail, and describes a number of mail systems and policies.

2 Sendmail

One of the more popular mail programs. Important to note that sendmail is *not* a mail delivery program but a central program which collects mail from different programs and then calls upon the appropriate mailer to do the actual delivery. Since sendmail is configured using a configuration file on startup, new mailers can be inserted at no extra cost. Some of the mailers used by sendmail are mail, uux, sh. It also acts like an SMTP mailer which is referred to as tcp & tcpld.

A tutorial was posted by Chuck Hedrick (of Rutgers University) several years ago, on the net.unix-wizards mailing group. One of the main points emphasized there was that sendmail *headers & envelope* are distinct and must not be confused.

Header refers to the lines at the beginning of the message, starting with "from:", "to:", "subject:", etc. Sendmail processes these lines under control of the startup configuration file. E.g. with uucp mail it will add its own host name at the beginning of the from line, so that the final recipient stands some change of replying to the message. However sendmail normally does not depend upon the From and To lines to perform its actual delivery. It has more direct knowledge, passed on to it from the program that generated the mail, or if it came from another site, the mailer at that site. This information is referred to as the "envelope", since it is like the addresses on the outside of an envelope. For Arpanet mail, the envelope is passed to the next site by the `MAIL FROM:` and `RCPT TO:` commands. For UUCP mail, it is passed on as arguments to the remote rmail command.

A small test was done to demonstrate the above. It is included in the appendices.

In the conventional, unmodified BSD sources for sendmail v5.65, one could not try to match a *class* which has a "dot" in it (because the "dot" is used as a delimiter. [1] Secondly, if defining a class of aliases for the hostname, the aliases get replaced by the hostname in all the headers.

At JvNCnet, the following approaches were considered:

1. To accept mail to `x.jvnc.net, x.csc.org, x.ieee.org` without replacing the `x.csc.org` with the real name `x.jvnc.net`, and accept the mail as local.

   This was achieved by defining E and F as new domains (notice that they are not a new class since couldn't match a class with a "dot" in them). Then, in the machine dependent part of ruleset #0,

   ```
   R$*<@$w.$E>            $@$>0$1            @host.aliasdomain ->
   R$*<@$=w.$E>           $@$>0$1            @hostclass.aliasdomain ->
   ```

2. To accept mail to anyuser@jvnc.net, and anyuser@csc.org (*i.e*, mail addressed to anyuser at our domain, or fake domain), without substituting the headers:

[1] Someone suggested removing the "dot" as a delimiter from the config file and the sources, but I preferred leaving it in till the Gods of Sendmail decided to take it out

```
## Accept mail to our domain - <user@domain>
R$*<$D>                 $@$>0$1                 user@domain ->
## Accept mail to our aliasdomain_E <user@domainE>
R$*<$E>                 $@$>0$1                 user@domainE ->
```

3 MMDF

4 X.400

5 MH

6 IMAP

7 POP

A SENDMAIL TESTS

These tests were done at JvNCnet, where the sendmail configuration file on the host bambi was modified to reflect all possible values in the headers. The version of sendmail used on bambi was 5.65 from BSD. None of the sources were modified.

```
#
# This is the portion of sendmail.cf file used on "bambi" to
# generate additional information for the tests following.
#
#

#########################
#   Format of headers   #
#########################

H?P?Return-Path: <$g>
HReceived: $?sfrom $s $.by $j ($v/$Z)
id $i; $b
Macros:
a: $a; b: $b; c: $c
d: $d; f: $f;
g: $g; h: $h
i: $i; m: $m; p: $p; r: $r
s: $s; t: $t; u: $u
v: $v; w: $w; x: $x ; z: $z
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $q
H?F?From: $q
H?x?Full-Name: $x
HSubject:
# HPosted-Date: $a
# H?l?Received-Date: $b
H?M?Resent-Message-Id: <$t.$i@$j>
H?M?Message-Id: <$t.$i@$j>



###########################
###   Rewriting Rules   ###
###########################
```

```
From aggarwal@nisc.jvnc.net Wed Nov  7 18:21:13 1990
Received: from nisc.jvnc.net by bambi.jvnc.net (5.65/1.34)
id AA08675; Wed, 7 Nov 90 18:21:12 -0500
Macros:
a: Wed, 7 Nov 90 18:21:15 -0500; b: Wed, 7 Nov 90 18:21:12 -0500; c: 0
d: Wed Nov  7 18:21:12 1990; f: aggarwal@nisc.jvnc.net;
g: aggarwal@nisc.jvnc.net; h:
i: AA08675; m: ; p: 8675; r: SMTP
s: nisc.jvnc.net; t: 9011072321; u: aggarwal
v: 5.65; w: bambi.jvnc.net; x:  ; z: /home/aggarwal
Received: by nisc.jvnc.net (5.61/1.34)
id AA26347; Wed, 7 Nov 90 18:21:15 -0500
Date: Wed, 7 Nov 90 18:21:15 -0500
From: aggarwal@nisc.jvnc.net (Vikas Aggarwal)
Message-Id: <9011072321.AA26347@nisc.jvnc.net>
To: aggarwal@bambi.jvnc.net
Subject: Test from NISC
Status: R


This is a test message sent from NISC.JVNC.nET to bambi with
the config file on bambi modified to reflect all possible
parameters in the mail headers. This is done as a realtime
tutorial on "sendmail".

The message has been sent using:

        mail -v aggarwal@bambi

Hostname on nisc returns "nisc.jvnc.net"
Hostname on bambi returns "bambi"


-vikas

  vikas@nisc.jvnc.net (609) 258-2403
-------------------------------------------------------------------------


From wrong!user@messed.domain Wed Nov  7 18:26:18 1990
Received: from nisc.jvnc.net by bambi.jvnc.net (5.65/1.34)
id AA08687; Wed, 7 Nov 90 18:24:29 -0500
Macros:
a: Wed, 7 Nov 90 18:24:29 -0500; b: Wed, 7 Nov 90 18:24:29 -0500; c: 0
d: Wed Nov  7 18:24:29 1990; f: wrong!user@messed.domain;
g: wrong!user@messed.domain; h:
i: AA08687; m: ; p: 8687; r: SMTP
s: nisc.jvnc.net; t: 9011072324; u: aggarwal
v: 5.65; w: bambi.jvnc.net; x:  ; z: /home/aggarwal
Date: Wed, 7 Nov 90 18:24:29 -0500
From: wrong!user@messed.domain
```

```
Message-Id: <9011072324.AA08687@bambi.jvnc.net>
Apparently-To: <aggarwal@bambi>
Status: R

This entire message is sent with the sendmail config file on bambi in
test mode and by telnetting to the sendmail port on bambi.

 nisc-v% telnet bambi 25
 Trying 128.121.50.149 ...
 Connected to bambi.jvnc.net.
 Escape character is '^]'.
 220 bambi.jvnc.net Sendmail 5.65/1.34 ready at Wed, 7 Nov 90 18:22:28 -0500
 HELO fake.domain
 250 bambi.jvnc.net Hello fake.domain (nisc.jvnc.net), pleased to meet you
 MAIL From:<wrong!user@messed.domain>
 250 <wrong!user@messed.domain>... Sender ok
 RCPT To:<aggarwal@bambi>
 250 <aggarwal@bambi>... Recipient ok
 DATA
 354 Enter mail, end with "." on a line by itself
 .

-vikas
----------------------------------------------------------------------

From aggarwal@bambi.jvnc.net Wed Nov  7 18:18:16 1990
Received: from jvncb.csc.org by nisc.jvnc.net (5.61/1.34)
id AA26336; Wed, 7 Nov 90 18:18:15 -0500
Received: from bambi.jvnc.net by jvncb.csc.org (5.61/1.34)
id AA22585; Wed, 7 Nov 90 18:18:22 -0500
Received: by bambi.jvnc.net (5.65/1.34)
id AA08671; Wed, 7 Nov 90 18:18:06 -0500
Macros:
a: Wed, 7 Nov 90 18:18:06 -0500; b: Wed, 7 Nov 90 18:18:06 -0500; c: 0
d: Wed Nov  7 18:18:06 1990; f: aggarwal;
g: aggarwal@bambi.jvnc.net; h: jvncb.csc.org
i: AA08671; m: ; p: 8671; r:
s: ; t: 9011072318; u: vikas%nisc.jvnc.net@jvncb.csc.org
v: 5.65; w: bambi.jvnc.net; x: Vikas Aggarwal ; z:
Date: Wed, 7 Nov 90 18:18:06 -0500
From: aggarwal@bambi.jvnc.net (Vikas Aggarwal)
Message-Id: <9011072318.AA08671@bambi.jvnc.net>
To: vikas@nisc.jvnc.net
Subject: Test From Bambi
Status: R


This is a test message sent from Bambi. The sendmail config
file on bambi has been changed to include all possible parameters
in the header. It has been routed through another host to see
the effect that it would have on the headers.
```

```
It has  been sent using:

mail vikas%nisc.jvnc.net@jvncb.csc.org


-vikas

  vikas@nisc.jvnc.net (609) 258-2403
--------------------------------------------------------------------------

From aggarwal@jvncb.csc.org Wed Nov  7 18:43:31 1990
Received: from bambi.jvnc.net by nisc.jvnc.net (5.61/1.34)
id AA26397; Wed, 7 Nov 90 18:43:30 -0500
Received: from jvncb.csc.org by bambi.jvnc.net (5.65/1.34)
id AA08743; Wed, 7 Nov 90 18:43:23 -0500
Macros:
a: Wed, 7 Nov 90 18:43:35 -0500; b: Wed, 7 Nov 90 18:43:23 -0500; c: 0
d: Wed Nov  7 18:43:23 1990; f: aggarwal@jvncb.csc.org;
g: aggarwal@jvncb.csc.org; h: nisc.jvnc.net
i: AA08743; m: ; p: 8743; r: SMTP
s: jvncb.csc.org; t: 9011072343; u: vikas@nisc.jvnc.net
v: 5.65; w: bambi.jvnc.net; x:  ; z:
Received: by jvncb.csc.org (5.61/1.34)
id AA22625; Wed, 7 Nov 90 18:43:35 -0500
Date: Wed, 7 Nov 90 18:43:35 -0500
From: aggarwal@jvncb.csc.org (Vikas Aggarwal)
Message-Id: <9011072343.AA22625@jvncb.csc.org>
To: vikas@nisc.jvnc.net
Subject: Test message
Status: R


This is a test message sent from jvncb.csc.org via bambi to
NISC. The config file on bambi has been modified to reflect
all sendmail parameters for learning.

This message was sent from JvNCB.csc.org using

mail vikas%nisc.jvnc.net@bambi.jvnc.net


-vikas

  vikas@nisc.jvnc.net (609) 258-2403
--------------------------------------------------------------------------


From aggarwal@jvncb.csc.org Wed Nov  7 18:45:18 1990
Received: from nisc.jvnc.net by bambi.jvnc.net (5.65/1.34)
id AA08747; Wed, 7 Nov 90 18:45:17 -0500
```

```
Macros:
a: Wed, 7 Nov 90 18:45:28 -0500; b: Wed, 7 Nov 90 18:45:17 -0500; c: 0
d: Wed Nov  7 18:45:17 1990; f: aggarwal@jvncb.csc.org;
g: aggarwal@jvncb.csc.org; h:
i: AA08747; m: ; p: 8747; r: SMTP
s: nisc.jvnc.net; t: 9011072345; u: aggarwal
v: 5.65; w: bambi.jvnc.net; x:  ; z: /home/aggarwal
Received: from jvncb.csc.org by nisc.jvnc.net (5.61/1.34)
id AA26403; Wed, 7 Nov 90 18:45:21 -0500
Received: by jvncb.csc.org (5.61/1.34)
id AA22630; Wed, 7 Nov 90 18:45:28 -0500
Date: Wed, 7 Nov 90 18:45:28 -0500
From: aggarwal@jvncb.csc.org (Vikas Aggarwal)
Message-Id: <9011072345.AA22630@jvncb.csc.org>
To: aggarwal@bambi.jvnc.net
Subject: Test for sendmail

The config file on bambi has been modified to reflect all sendmail
parameters. This mail has been sent from jvncb.csc.org to bambi
using:

mail aggarwal%bambi@nisc.jvnc.net

Note that bambi is in the jvnc.net domain.


-vikas

  vikas@nisc.jvnc.net (609) 258-2403
--------------------------------------------------------------------------
```

B SENDMAIL MISC INFO

```
Path: jvnc.net!princeton!att!linac!mp.cs.niu.edu!rickert
From: rickert@mp.cs.niu.edu (Neil Rickert)
Newsgroups: comp.mail.sendmail
Subject: Re: List of separators in sendmail
Message-ID: <1991Apr26.132742.18354@mp.cs.niu.edu>
Date: 26 Apr 91 13:27:42 GMT
References: <1991Apr26.045337.10580@jvnc.net>
Organization: Northern Illinois University
Lines: 27

In article <1991Apr26.045337.10580@jvnc.net> vikas@jvnc.net writes:
>I am running the latest version of sendmail from Berkeley on a couple of
>Suns.

 Both the IDA versions of sendmail and your Sun supplied 'sendmail' support
multi-token class matches.

>My question is  this  -  "Can   I safely remove   the '.' from  the  list of
>separators in the config file (Do.:%@!^=/[]) ??"

 If you do this you lose all ability to recognize address such as node.BITNET
or node.UUCP and treat them differently from mail with Internet addresses.  You
lose a great deal of flexibility.

>P.S. I remember seeing  something on the  net about the  Ida sendmail mailer
>where  they had  mentioned  that  the '.' in   the list of  separators was a
>carryover from teh old Berknet days and should be removed.

 This was a comment about addresses of the form x.y without any '@' in it
at all.  It had nothing to do with the '.' within domain specifications.


--
=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=*=
  Neil W. Rickert, Computer Science               <rickert@cs.niu.edu>
  Northern Illinois Univ.
  DeKalb, IL 60115                                   +1-815-753-6940
```

C MAIL SITE CONFIGURATIONS

This section is an excerpt from the tcp-ip mailing list which discussed the different mailing strategies used at a number of sites.

```
From tcp-ip-RELAY@SRI-NIC.ARPA Tue May 23 17:59:16 1989
Received-Date: Tue, 23 May 89 17:59:11 EDT
Received: by jvncb.csc.org (5.52.1/6.13)
id AA21756; Tue, 23 May 89 16:59:52 EDT
Received: by ucbvax.Berkeley.EDU (5.61/1.36)
id AA26927; Tue, 23 May 89 10:01:06 -0700
Received: from USENET by ucbvax.Berkeley.EDU with netnews
for tcp-ip@sri-nic.arpa (tcp-ip@sri-nic.arpa)
(contact usenet@ucbvax.Berkeley.EDU if you have questions)
Date: 23 May 89 16:59:51 GMT
From: cheops.cis.ohio-state.edu!pritch@tut.cis.ohio-state.edu  (Norm Pritchett)
Subject: Centralized mail systems summary (LONG)
Message-Id: <49488@tut.cis.ohio-state.edu>
Sender: tcp-ip-relay@sri-nic.arpa
To: tcp-ip@sri-nic.arpa
Status: RO
```

A few weeks ago I posted a query to find out what people are doing with centralized mail systems. I promised to followup with a summary of the responses and to let you know what we were going to do at Ohio State. The latter will be in my next posting. Below are a summary of responses. I will mention a contact name for each of the mail systems listed below but it might not be a person involved with that project - merely a user who described the mail system to me.

Sun Microsystems Of the ones described to me this appears to be the most well known – I received messages from 5 people plus Sun about it. Contact: Bill Melohn ¡melohn@eng.sun.com¿.

At Sun we have managed to do this for our some 10,000 employees. We currently use a two-tier method of resolving names from a unix user name (like "melohn") to ¡first initial of first name¿¡lastname¿ (ie "bmelohn") syntax. This alias in turns points to the username@mailboxhost for the user. When conflicts occur in the scheme, we make the alias a pipe to a shell script called ambigmail, which sends mail back to the sender with the various GCOS field entries that match the ambig alias.

We are in the process of enhancing this scheme by making the second alias from above reference a username@Area, which would allow us to distribute the alias expansion to a series of mailhosts for each area.

For mail destined for the Internet, we rewrite outgoing mail headers to be user@Sun.COM; eventually this function will be done on an area basis, with outgoing mail messages that look like user@Area.Sun.COM (as mine does today).

UC Davis Contact: jcgargano@ucdavis.edu (Joan Gargano).

I am in charge of our mailname system at U.C. Davis. We have about 20,000 faculty and staff, and 20,000 students. I maintain a database of over 20,000 mailnames, first initial, middle initial, last name, for faculty and staff which I constructed from the payroll files. We have had a number of collisions which I have resolved by altering the middle initial of one of the names. Stanford uses a similar system.

You may query our database by using whois:

There is a directory that is accessible via the whois program. We have added to the whois program to search our local database through the program or through electronic mail.

Purdue University Contact Dave Stevens (dls@alecto.cc.purdue.edu) .

We've just started work on what we call campus-wide electronic mail. We intend to use name server MR records and Profile for the white pages service. We're thinking about using NeXT work stations in a distributed model.

Proteon Inc. Contact Alan Marshall ¡acm@proteon.com¿.

I can recommend CCMail as the product to work with pc networks. This is connectable to SMTP mail via a public domain translator that I have written. It is available from monk.protoen.com as arpagw.arc in the /ftp/pub directory. There is another implementation done from my implementation by Mike Morse (mmorse@nsf.com) that uses about 10k users in the system. He would be a good one to talk with about your needs. I have some code from him too and he has said to make it available. Perhaps there is a more current version that would help.

MIT Contact Mark Rosentein ¡mar@athena.mit.edu¿.

We've been thinking of tackling this problem here at MIT. Our initial planning is as follows:

* The full name of every member of the MIT community will be known to the mail hub. Mail sent to someone's full name will result in:

1) The mail is delivered if the name is unique and the person has a mailbox 2) An error response is generated saying "[full name] does not have an electronic mail address, please send mail to MIT Room ..., Cambridge MA 02139" 3) An error response is generated saying "[full name] is ambiguous, please choose one:" followed by a list of people giving the name, title, address, and a unique email identifier. 4) An error response saying "addressee unknown".

* Every member of the MIT community will be given a unique identifier for email purposes. For most active email users, this will be their login name. For other people and those with name conflicts, it will be their initials and a number, similar to the NIC's whois database.

This information will be kept up-to-date by Moira, the Athena Service Management System, and regularly updated on the mailhub. Users will be allowed to update some of their own information, and to become unlisted if they want to.

Moira currently contains all of the necessary information for the students here at MIT, only the staff and remaining faculty must be added. The primary development effort will be modifications to the mail hub.

NCR Contact Matt Costello ¡Matt.Costello@sandiego.nr.com¿

Well, I can help out some on this. I designed the system in use throughout NCR and it conforms to your requirements. There are 1̃300 people with email addresses in San Diego. I believe Dayton (Galactic Headquarters) has around 6000 email addresses in it.

What I did was to create a database external to the mail system and then have the mail router look up certain addresses in this external database. Any mail addressed to the domain name, or having a period in the username will be looked up in this database. The database format is a rolodex(tm) format. My entry is simply

```
name Matthew Costello
phone 2926
```

```
dept 4796
email mattc@ncr-sd
```

This database format is simple to manipulate and edit using the standard unix tools, but there are also 7 programs (rolo, roloeach, roloedit, roloenter, rolorev, rolorpt & rolosort) that handle it more efficiently. The command rolo(1) is used to look up the local portion of the name using a fuzzy matching technique, so the following will all find my entry and get my mail to me:

```
matthew.costello
matt.costello
costello I'm the only Costello in San Diego
pat.costello
m.castelli
```

[Accepting the last two was a mistake. It would be better to fail and then [return the close matches.

Because of the fuzzy matching the search must be linear through the whole file. To compensate our mail router is able to cache found addresses in a separate file so they only get looked up once. I would recommend using an "initial substring match" which is amenable to indexing.

Universite Catholique de Louvain Contact Alain FONTAINE ¡FNTA80@BUCLLN11.BITNET¿.

We have established an unified address scheme here. But we did not find any way to allow external correspondants to send mail to an individual when only knowing his name, *and* avoid clashes... This seems theoretically impossible. The sender must *know* and *specify* some more information to garantee uniqueness. So the addresses used are of the form : personal-identifier@unit.ucl.ac.be, where 'unit' is the standardized three or four letter sigle of the laboratory or service in which the person can be found. Of course, it is difficult for an external correspondant trying to contact somebody for the first time to guess the 'unit' to be used. On the other hand, clashes are a very low probability event, since units never count more than 50 persons.

Implementation : the DNS would be a marvelous tool for this, since each unit could have and manage its own name server. Halas, (one of my favorite gripes), the arbitrary division of mail addresses into a local and a domain part makes it impossible to use the DNS down to the individual level. So the current situation is that one centralized machine contains a centralized database of mail routing information, and nearly all domain-addressed mail goes physically (uh, should we say that about zeroes-and-ones on wires and disks and ...) through that machine.

Carnegie-Mellon University Contact Craig Everhart ¡cfe+@andrew.cmu.edu¿.

Andrew supports 8500 user names reasonably gracefully, though we've given up on making login-names guessable; too many collisions. Instead, we use a White Pages service to map name probes to mailboxes, letting it handle any collisions.

My free advice to you would be to forget making names unique; they never will be. Make login-names unique and provide simple ways to map from person-names to login-names (and use them for delivering incoming mail). MCI Mail, with a cast of many hundred thousand, did the same thing; everybody's mailbox is a number.

University of Illinois at Urbana Contact Paul Pomes ¡Paul@uxc.cso.uiuc.edu¿.

The Computing Services Office at the University of Illinois at Urbana is in the process of creating a university-wide mailing system. The system is comprised of three pieces. The largest is the white-pages system created by Steve Dorner of CSO. It's based on the CSnet central name server (qi - Query Interpreter). Each student and staff member is assigned a unique alias. The user is allowed to change the issued alias provided it remains unique. Associated with this alias is the user's preferred email address, office address, home address, phone numbers, etc. Everything that is in the paper phone book is also in the qi database.

The user client is a program called ph. It searches on the unique alias and can fuzzy match on names. Providing ancillary information such as department or curriculum narrows the search.

The second piece is the 5.61+IDA sendmail release. The ida/cf/Sendmail.mc has been very slightly modified to invoke a new mailer, phquery, whenever an address resolves to ¡name¿@uiuc.edu. This is configured with the DOMAINMASTER option.

Phquery is the third piece. It examines its arguments and calls qi to determine the preferred email address for the supplied name. At this point, name can be only the unique qi alias. This restriction will soon be lifted to allow phquery to resolve full names (e.g., paul-pomes@uiuc.edu -¿ paul@uxc.cso.uiuc.edu), and amateur radio callsigns (e.g., ka9wgn@uiuc.edu -¿ phil@vmd.cso.uiuc.edu). In the case of ambiguous matches, phquery will return a list of possibilities that includes department and/or curriculum information that should allow the sender to make the next attempt successful.

Future enhancements include automated printing and campus mailing of messages to those users w.o. email addresses.

Source for the qi (central server) and ph (user client) can be obtained via anon-FTP from uxc.cso.uiuc.edu:/net/{ph,qi}. The phquery code, when ready, will be included in the /mail/sendmail/uiuc directory.

Sorry, we cannot email this code as it is much too large. Chocolate chip cookies with a postpaid tape will work wonders though.

University of Virgina Contact Tom Sigmon ¡tms@virgina.edu¿

I am responding to the request in BIG-LAN regarding university-wide electronic mail networks. We here at the University of Virginia have created such an environment that addresses most of the points that were brought up. Our electronic mail environment currently encompasses over 300 machines (not PCs, etc.) having many different mailers running on many different operating systems. I'll try to summarize the basic points here and if there are follow-up questions, I'd be happy to address them.

- we use domain addresses only. If a user wants to send mail to someone on a non-domain network, then they must use an appropriate "pseudo-domain" within a domain address. For example, sending mail to someone on Bitnet would require an address of the form "user@host.bitnet". Likewise, sending mail to someone on a UUCP host requires an address of the form "user@host.uucp" (our mailers figure out the best path to the target host).

- third-level domains within the "virginia.edu" domain are named after departments or other University organizations (usually using the standard registrar's designation)

- departments create whatever fourth-level domains or machine names (the usual case) that they desire

- we here in the Academic Computing Center created and maintain a database of every faculty, staff, or student associated with the University. The basic data comes from the registrar's database and the payroll database from our administrative computing center.

- as part of this database, we automatically create unique mail ids for every single person associated with the University. These ids are also (conveniently) used as the login id on most machines. The format of this unique id is as follows: person's initials optionally followed by a 2-character suffix whose first character is a digit and whose second character is alphabetic. For instance, my mail/login id is simply my initials, "tms". All other people who have the same initials as me have a suffix on their mail id, e.g., tms2x, tms4g, etc. Obviously, the choice of format and a priori creation of unique mail/login ids is the most controversial part of our environment. There are advantages and disadvantages of this system which I won't go into unless someone is interested.

- since all of these mail ids are unique, they can be considered to be "aliases" in the "virginia.edu" domain. We support the notion of "registration" which creates a mapping between a person's unique mail id (in the virginia.edu domain) and the actual account and domain where that person reads his mail. For example, all mail sent to "tms@virginia.edu" will be delivered to the place where I actually read my mail which is "tms@boole.acc.virginia.edu". Thus, no one needs to know the details of exactly where I read my mail. Every system in our environment allows users to set/change their registration since it is done via a mail message to one of our main mail servers. Most systems wrap a shell script around this registration process so that it is very easy for the user to register or make changes.

- the above "registration" process is very important for mail coming to users from networks that don't support domain names (e.g., Bitnet and UUCPnet) as well as to present one "name" for the University to the outside world. In these cases, if a user is not registered, then our mail servers would not know where to actually deliver the person's mail *especially* since we want to present one "name" to the outside world (i.e., it shouldn't be necessary for anyone to know the full domain name in order to send mail to someone at the University, nor should they need to know the internal network configurations, etc.). For example, the University has one name/address on both Bitnet and UUCPnet. We are "virginia" on both networks, so someone on Bitnet can send mail to me as "tms@virginia" without regard to the actual machine I use to read my mail, and likewise, someone on UUCPnet can send mail to me as "...!virginia!tms" without regard to the actual machine I use to read my mail.

- we also support user-created aliases at the virginia.edu level. If a user does not like their automatically created unique mail id or would simply prefer to have other aliases, then they can request the creation of such aliases. For example, in addition to sending mail to me as "tms@virginia.edu", people can also send mail to me as "sigmon@virginia.edu" or "9240615@virginia.edu" (which is my phone number). The only restriction that we place on these user-created aliases is that they (obviously) must be unique, can not conflict with the regular expressions that describe our automatically-generated ids (so that they don't preempt future automatically-generated ids), and that they be "reasonable" (e.g., we don't allow people to be MickeyMouse, nor GeorgeBush, etc.).

- of course, none of the above prohibits users from having aliases in other domains. The Academic Computing Center administers ids and aliases at the virginia.edu level for the entire University Departments are free to have their own aliases in their own domains (except that mail coming from non-domain-based networks can't access them for obvious reasons).

- the University telephone book has a section that lists the electronic mail ids and aliases for all registered mail users at the University. In addition, we support a "whois" capability on many of our machines that allows users to interactively query our database to determine mail ids, phone numbers, department affiliations, etc.

Hope this helps others establish university-wide mail networks. I'm happy to provide more detail or answer questions, just send me mail!

Stanford University Contact Bob Morgan ¡morgan@jessica.stanford.edu¿.

Yes, assigning what we have come to call a "unique-id" to a campus-full of people is a tricky issue. We have made a few abortive attempts at campus-wide mail delivery (unique-id@stanford.edu), but have run into the twin problems of a) choosing the unique-id, and b) letting people update their unique-id/actual-mailbox mapping without involving great piles of paper/bureaucracy/our time.

Right now we generate unique-ids for use with a phone-book-type service (based on Whois, RFC 912), using the following algorithm, moving down the list in case of name clash:

```
1) first-initial/last-name (rmorgan), or
2) first-initial/middle-initial/last-name (rlmorgan), or
3) as-many-initials-as-necessary/last-name (rlfmorgan), or
```

```
4) as-many-letters-of-first-name-as-necessary/last-name (robmorgan),
or
5) first-name/middle-initial/last-name (robertlmorgan), or
6) 5) with digits as necessary appended (robertlmorgan3).

(If you have a Unix ''whois'' client, you can bang our server with:
  > whois -h argus.stanford.edu some-string)
```

Looking through our 153 Smiths, I see no uses of rule #6, about 5 cases where #5 was used, and several instances of repeated application of #4 (dasmith, davsmith, davismith, davidsmith). I suspect that if we started using this for actual mail delivery (or, even more so, for Kerberos-style principal ids), some people would complain and insist on choosing their own. The question then becomes, how do you decide what's reasonable? If Joe Student wants to be known as "donaldkennedy" (SU's president), is that OK? Part of the problem is that if these things are assigned immediately when people arrive (as they must be) then people will be stuck with something before they know what it's about (as with real names, I suppose).

No solutions, just more questions,

Dartmouth University Contact Steve Campbell ¡Steve.Campbell@dartmouth.edu¿.

Dartmouth has a scheme much like the one you describe, called the Dartmouth Name Directory. The DND is a database of about 13,000 names with corresponding nicknames, password, paper-mail address, phone number, department (or undergraduate class), and e-mail address. Mail addressed to Joe.Blow@dartmouth.edu goes to Joe Blow's preferred e-mail address, as it is recorded in the DND.

People are uniquely identified by the tokens in their name – the name space is small enough that first name + middle initial + last name is unique in all but a very few cases. Those people have their middle names entered also. The names, nicknames, and departments are all lookup keys and partial matches are supported. So you can mail to me with "James.W.Matthews@dartmouth" (my full name), "James W M" (abbrieviating the last name) or "Jim Matthews" (matching my last name and a nickname). Only one token match must be exact.

If there are multiple matches a bounce message is generated, listing the matches (as long as there are fewer than fifteen or so). So it is fairly easy to refine an address to the required precision.

The DND is seeded by the personnel and registration systems, and several fields (paper mail address, e-mail address, phone number, and nicknames) are user-maintained. The default e-mail address is our paper mail system – messages are printed out and hand delivered.

UCS Contact Brian Kantor ¡brian@ucsd.edu¿.

UCSD has such a mail system. You may query it over the net to see what it looks like with the 'whois' command; try

```
whois -h ucsd.edu smith
whois -h ucsd.edu jsmith
```

and variations along that line.

The software to implement this is available in our anonymous FTP directory; take file pub/mailreg.tar.Z. Caveat Emptor: the software is continuously being refined and is not documented.

University of Kent at Canterbury Contact ¡sjl@ukc.ac.uk¿.

We operate an unified mail system for some 4000 staff and students. We use a centralised admin server which allocates a unique userid for each user. In addition it will allocate a login (also unique) for the machines they require.

On top of the admin server we run a mail database system (original designed at Edinburgh University). The user interface to this is the mailhost program. the user may nominate any machine he can log into as his mail machine. He does this by typing “mailhost -c”. At night all the machines swap lists of users. Each entry in the list has a date stamp, if this stamp is later than the local machines recorded gate stamp for the user the entry is updated.

An extension to this is being worked on at the moment which allows psuedo domains. A group of workstations (typically) have a pseudo domain centered on their fileserver. This is still experimental.

The system has been in use for about three years now with no problems.

UMCP Contact Mark Feldman ¡feldman@umd5.umd.edu¿.

I know of a few places that do this sort of thing. First, there’s the UMAIL project here at UMCP. One can send mail to various user names at host umail.umd.edu, and the mail gets delivered, even if the user doesn’t have a computer account anywhere. (In that case, mail gets printed out and sent via campus mail.) I am by no means fully up on the details of this system; you might talk to Mark Feldman (feldman@umd5.umd.edu) to get more information. He may not be the right contact, but he can probably name the right people if asked.

...

Contact Steve ¡steve@umiacs.umd.edu¿

Third, I’m implementing something that, while it doesn’t do everything, does most of what you want (and which can be extended to do more). Basically, it’s an automatic method of generating a ’global’ mail address database, made from the union of the password and alias files for a particular department. Getting all such files for a whole campus would be hard, but there’s also a facility for getting just another department’s global database and merging it in with others. There’s a concept of locality-of-reference, too; if I send mail to ’root’, I get the UMIACS root, even though there’s a ’root’ in the CS Department’s imported information. This locality can be guaranteed either manually or automatically.

If extended with some software to generate full-name addresses (First.MI.Last), and some software to handle duplicates differently, my code would probably do everything you need. The only hitch is that the stuff I’ve written is not yet in extensive use (so I’d bet it has, er, misfeatures), and it’s essentially undocumented. If you want a copy of the code as it stands, I could provide one. It’s solid (it’s even been Saberized), but it’s definitely in need of some major cleaning up...

ATT Private Mail Exchange Contact Rod Hart ¡hart@cp1.cp.bell-atl.com¿.

Look into the AT&T Private Mail Exchange System (PMX). My organization is in the process of installing one right now to solve a similar problem. We need X.400 in order to tie the various user groups (ie. DG, Proffs, Wang, PC, and of course Unix) together as well as document conversion.

```
Norm Pritchett, The Ohio State University College of Engineering Network
Internet: pritchett@eng.ohio-state.edu BITNET: TS1703 at OHSTVMA
UUCP: pritch@sydney.columbus.oh.us CCNET: ENG::PRITCHETT (6172::PRITCHETT)
```

D Miscellanous Mail Information

```
From owner-sun-managers@nisc.jvnc.net Fri Feb 15 00:35:32 1991
Date: Thu, 14 Feb 91 18:19:11 PST
From: d3a497%socrates.pnl.gov@pnlg.pnl.gov
Subject: Re: VMS .vs. UNIX mail hosts
To: sun-managers@eecs.nwu.edu, sun-nets@umiacs.umd.edu
Message-Id: <9102150219.AA00334@socrates.pnl.gov>
X-Envelope-To: sun-managers@eecs.nwu.edu
Status: R


Thanks to all those that replied to my Request For Info (RFI)...
Here is my original posting followed by a summary of the responses.



Our site is presently considering the option of moving our mail host from
a VAX/VMS to a UNIX platform.  We presently are using a VAX 8350 running VMS
as our one an only Internet gateway.  The system uses PMDF as a mailer  and
acts as our sites mailhost to the Internet.  This system also provides our site
with VMS <=> UNIX mail translation.

We also have a LAN (4000 nodes) which use CC-mail, the e-mail site standard
for PC's and Mac's.  We would like to find a system that will accept UNIX mail
to/from the Internet and forward it to/from one of our:

1) UNIX systems
2) VAX/VMS systems
3) PC's or Mac's via CC-mail

1) Are there others out there that have transferred their site's mail host from
   a VMS to a UNIX system?  What are the possibilities, concerns and traps that
   we need to take into consideration?

2) Is there anyone who has found a SMTP <=> cc-mail gateway that works.  We
   have tried cc-mail's gateway but have not had much success.

The system must support:

  1) Domain Name Service (DNS)
  2) RFC's 733, 822 and 1127
  3) Mail Exchange (MX)
  4) 10,000 mail aliases
  5) mail queueing
  6) mail error logging
  7) access to BitNet.


Troy Thompson
Battelle NW Labs
```

Richland, WA. 99337

Internet Address: tk_thompson@pnl.gov

HERE ARE THE VARIOUS RESPONSES

>From rusty@belch.Berkeley.EDU

Have a look at zmailer. It's available via anonymous
ftp from cc.utoronto.edu. It's a replacement for the
unix sendmail mail daemon.

>From wrl%wdl51@wdl1.wdl.loral.com

Hi,
I'm not sure about the CC-mail relay but you could use a DECstation 3100
running ULTRIX w/ ULTRIX-DECNet software.

The ULTRIX provides DNS w/MX support (your 10,000 aliases is quite large
but I guess it will work). The ULTRIX-DECNet provides Telnet, FTP, and
SMTP translation to VMS/DECNet.

We have two ULTRIX systems with DECNet on it. They allow TCP users
to access VMS systems and vice versa. Also the mail is relayed from
SMTP to VMS Mail and vice versa. We use both machines as mail
relays. All our DECNet systems across the U.S. are known as
"*.dnet.loral.com". There is an MX record that points to either
"wdl72.wdl.loral.com" or "ssd1.ssd.loral.com". To the Intenet user
or the DECNet user it looks like the mail came through the same gateway.
We even modify the return address so it look like it came from
"user@sjecc1.dnet.loral.com" instead of "user%sjecc1.dnet@wdl72.wdl.loral.com"
(which is the standard formatr from DEC).

The other nice feature of ultrix is that you can specify which type
of address look-up you want. You tell your system to check the
/etc/hosts or DNS first for address resolution. We have a small host
table in /etc/hosts. This allows the boot to go smoother. If its
not in the local tables, the DNS is used.

Hope this helps,

Bill Lewandowski
Loral WDL

```
---------------------------------------------------------------------------
>From mstar!root@uunet.UU.NET

I've never heard of CC-mail, but I suspect that Sendmail can be
coerced into handling all your requirements.  Otherwise, there are
MMDF implementations for UNIX, which you might even prefer because of
prior familiarity.


---------------------------------------------------------------------------
>From jim@math.psu.edu


I suggest you look into the IDA sendmail kit (available via anonymous ftp
from uxc.cso.uiuc.edu).  It's full of features, and very flexible.  Among
other things, it has built-in support for mail11v3 for DECnet, which will
probably help your connectivity to your VMS machines.

It supports dbm lookups directly via the configuration file, so your number
of aliases is not a problem.  This also supports generic e-mail names, so
you can continue to remap names like tk_thompson@pnl.gov to d3a497%socrates
on incoming mail, and also have it remapped on outgoing mail as well.

Since it's sendmail, it does DNS, the RFC's, MX, and queuing and logging.

If you're interested in BITNET connectivity on your mail gateway running
Unix, then you should look into UREP, the Unix RSCS Emulation Package.  This
provides the necessary functions to connect Unix boxes to BITNET.  It's
available for a small license fee from Penn State Computer Science.  If you
want more info on UREP, send mail to urep-info@cs.psu.edu.

Jim


---------------------------------------------------------------------------
>From mmorse@NSF.GOV

>  We would like to find a system that will accept UNIX mail
> to/from the Internet and forward it to/from one of our:
>
>  1) UNIX systems
>  2) VAX/VMS systems
>  3) PC's or Mac's via CC-mail

> 2) Is there anyone who has found a SMTP <=> cc-mail gateway that works.  We
>    have tried cc-mail's gateway but have not had much success.

Our's works.  If you like, I'll FAX you a copy of my report on
cc:Mail's SMTP gateway that explains some of the problems they have
that we don't.  cc:Mail has the report, and has said they will address
```

```
the problems.

> The system must support:
>
>   1) Domain Name Service (DNS)
>   2) RFC's 733, 822 and 1127
>   3) Mail Exchange (MX)
>   4) 10,000 mail aliases
>   5) mail queueing
>   6) mail error logging
>   7) access to BitNet.

Our system supports these easily (except what is RFC 1127?).  Our mail
hub is an elderly VAX 785 running Ultrix and MMDF.  We would like to
replace it with a more powerful machine, but budget concerns, etc.,
have slowed this project down.  We run UREP with a direct RSCS
connection to BITNET, which we are replacing (again, slower than I'd
like) with a BITNET II (RSCS over TCP/IP) connection.  We use PMDF to
exchange mail with an HP system and for the cc:Mail gateway.  About 400
of our users log on to the VAX, and the other 800 use cc:Mail.  While
our user population sounds quite a bit smaller than yours, I think our
system would scale up to yours easily, given the right size hardware.

Please feel free to write or call for details.

--Mike


---------------------------------------------------------------------------
>From berg@pwtc.tc.pw.com

> 2) Is there anyone who has found a SMTP <=> cc-mail gateway that works.  We
>    have tried cc-mail's gateway but have not had much success.

We've more or less built our own SMTP to cc:Mail gateway.  We've been
running cc:Mail on our PCs for several years (we were a very early
customer of cc:Mail), and at that time they didn't have an SMTP gateway.
Instead, we used their IMPORT/EXPORT programs.

We have a PC which is connected both to our PC LAN (running Novell
NetWare) and to our TCP/IP/NFS network (running Sun's PC-NFS).  We did
this by putting two network interface cards in the PC: PC-NFS runs over
Ethernet, while Netware runs over Token Ring.  Of course, there are other
ways to do the same thing.  (In fact, Novell is announcing their
TCP/IP/NFS NetWare Loadable Module at NetWorld this week, which would make
the "two network interfaces" hack unncessary.)

The heart of our gateway is a set of shell scripts (and related changes to
our sendmail.cf file) which take mail which is addressed to
"user@postoffice.ccmail.pw.com" and modifies the message so it is in
cc:Mail IMPORT format.  As part of this, the shell script re-write the
```

headers so that the address is "user AT POSTOFFICE" (and changes the "From" line as well). The file then gets put into a directory which is exported via NFS to the PC, which sits in a loop running IMPORT and EXPORT (and also GATEWAY, which we use to communicate via modem with other cc:Mail post offices). The PC imports that file into our cc:Mail post office.

For mail from cc:Mail to Unix users, things work very similarly. We defined a post-office called "Internet", and the EXPORT program takes any mail addressed to that post office and exports it to a file. That file is then processed by another set of shell scripts (running under cron) which tranforms the messages from cc:Mail EXPORT format into SMTP (actually, RFC 822) format, and passes them to sendmail for delivery.

The actual cc:Mail post office files live on the Novell file-server, so they are accessable to PCs on the LAN which don't use PC-NFS.

A member of our staff, Chris Burgess, has done a lot of work on this in the last few months, and we're pretty happy with the results. We used to have it set up so that each cc:Mail user needed an alias in /etc/aliases to be able to get mail from the Unix side, but that was just too much of a hassle to maintain. Setting up "ccmail" as a domain is a much more general approach.

By the way, we did look at the cc:Mail SMTP gateway product when it came out, and decided that it suffered too many limitations to be useful to us. (It seemed like it still would have required us to maintain Unix-side aliases for each cc:Mail users, which was what we were trying to avoid.) I'd be interested to hear what your experience was.

(In case you haven't heard, cc:Mail just got bought by Lotus; it was in this morning's papers. Since we also use Lotus Notes here, we're hoping that Lotus will fix their gateway to cc:Mail and make our lives a little easier!)

>From ardent!harker@uunet.UU.NET

As for the UNIX to VMS mail gateway, the best solution is to get a mail gateway (machine) from the company with the most interest in making the gateway work. That would be a Dec Ultrix box running DECnet. I used a lowly microvax for one and it worked like a charm. (only good use of a microvax that I know of)

Hope this helps
RLH

>From schwartz@groucho.cs.psu.edu

| I suggest you look into the IDA sendmail kit

As Jim says, IDA is the C++ of Sendmail. If creeping featurism
bothers you, you may want to consider Zmailer, by Rayan Zacharaissen
<rayan@ai.toronto.edu>, a very modular and extensible design.

>From jqj@duff.uoregon.edu

>2) Is there anyone who has found a SMTP <=> cc-mail gateway that works. We
> have tried cc-mail's gateway but have not had much success.

We have a similar mix of systems and requirements. We're running Al
Marshall's public domain version of the cc:mail gateway code, and it works
reasonably well. Big advantage is that since we have sources we can fix
it if we don't like it. cc:mail's product just isn't supportable.

We currently have about 400 cc:Mail mailboxes on 7 or 8 Netware file
servers. The gateway runs on 4 PCs of various flavors (I think 2 old PCs
and 2 AT clones at the moment).

>From lyndon@aupair.cs.athabascau.ca

> As Jim says, IDA is the C++ of Sendmail. If creeping featurism
> bothers you, you may want to consider Zmailer, by Rayan Zacharaissen
> <rayan@ai.toronto.edu>, a very modular and extensible design.

Or take a look at smail3.1 (available via ftp from ftp.uu.net. I
haven't seen zmailer yet so I don't know how it compares to smail3.1.
Either one HAS to be better than sendmail though ...

From news@tigger.jvnc.net Thu Dec 12 23:00:24 1991
Received: from tigger.jvnc.net by bambi.jvnc.net (5.65/1.34)
id AA09520; Thu, 12 Dec 91 23:00:20 -0500
Received: by tigger.jvnc.net id AA14849
(5.65c/IDA-1.4.4 for vikas@bambi.jvnc.net); Thu, 12 Dec 1991 23:00:15 -0500
Date: Thu, 12 Dec 1991 23:00:15 -0500
From: Zee News Genie <news@tigger.jvnc.net>
Message-Id: <199112130400.AA14849@tigger.jvnc.net>
To: vikas@bambi.jvnc.net
Subject: Network News Batch (121291,23:00)

Xref: jvnc.net news.admin:8932 comp.mail.misc:3744
Path: jvnc.net!yale.edu!qt.cs.utexas.edu!...!ecicrl!clewis
From: clewis@ferret.ocunix.on.ca (Chris Lewis)
Newsgroups: news.admin,comp.mail.misc
Subject: Draft 2 of Email Software FAQ
Message-ID: <2881@ecicrl.ocunix.on.ca>
Date: 12 Dec 91 02:09:27 GMT
Followup-To: news.admin
Organization: Elegant Communications Inc., Ottawa, Canada
Lines: 812

[This is draft 2. Please indicate draft number if commenting.
Changes are sufficiently general that you should read the whole thing.
It is rather large. I don't think too large tho.

This is a start on a FAQ for how to set up UNIX email systems.
Written more as a guide rather than specific questions. If you have
comments, please respond to me directly by email.

At present, this is only covering UNIX email - it's big enough as it
is.

Thanks to all of the people who gave me package reviews, and the
others on draft 1 in general.

If you're an author of email software, and you've not sent me a
review, PLEASE make sure I'm not lying about your offering;-)]

UNIX EMail Software - a Survey

Configuring electronic mail systems can be quite a complicated
subject. Often far more complicated than, say, setting up a USENET
news feed. This is because, unlike news, email is expected to
traverse multiple types of networks using their own protocol, whereas,
USENET news tends to be a single protocol supported by hook or by
crook on different networks.

This document is intended for the system administrator who needs to know how to set up their UNIX system for email communication with the outside world. It is intended for the email-naive SA (much like myself ;-) who gets more than a little confused by the acronyms, RFC's and plethora of software.

This is intended to be a general survey of the software available, so I won't spend too much time on some of the details. Most of the available software comes with documentation that can explain things much better than I can.

Additional detail can be obtained from several sources, such as:
"The Matrix", John Quarterman
"Using UNIX mail", Nutshell handbooks
[Could someone give me more complete references? Thanks]

Further, this is primarily oriented towards UNIX email systems. This is unfortunate, because it would be nice to have a general document covering email in all of its forms. However, each operating system tends to have radically different email mechanisms, so it would be difficult to do justice to any other environment. It seems more useful to cover one environment well here, and have companion documents for other environments. [Tho, if I get enough data on other systems, I might include it]

The first section is an overview of how mail systems go together, and a glossary of the important terms to know.

The second section is a discussion of the general do's and don'ts of mail systems, that lead into the third section which discusses which software you need.

The fourth section includes general overviews of some of the software available.

The more knowledgable reader will obviously note that this is being rather simplistic, but I feel that the really fine details are far beyond the scope of this document as well as the needs of most System Administrators. Many issues, such as configuring modems etc., are also beyond the scope of this document.

Electronic mail - overview and glossary:

Electronic mail generally consists of three basic pieces:
1) the link level transport - which could be UUCP, TCP/IP, or a host of others. We'll call this the "transport medium" (TM)
2) the "Transport Agent" (TA) which is responsible for transporting mail from source to destination, possibly transforming protocols, addresses, and routing the mail. Sometimes called a Mail Transport Agent (MTA)

The TA often has several components:
- Routing mechanisms
- Local delivery agent
- Remote delivery agent

Many TA's have all of these components, but some do not. In other cases, it is possible to replace certain components for increased functionality.

3) The "User Agent" (UA) is the user interface - the software that the user uses to read his mail, sort things around in folders, and send mail. Sometimes called "Mail User Agent" (MUA).

Perhaps the most important distinction between mail systems is that of the transport used - it tends to determine much of the rest of the software needed.

UUCP:

Back in the mists of time, UNIX systems communicated only over RS232 serial lines, usually over modems. UUCP is a suite of programs developed back in the early 70's to provide this communications link. All that UUCP does is transfer files from one system to another. There is an additional mechanism where one system can direct the destination system to run a file through a specific program. Electronic mail in UUCP is simply requesting the destination machine to run "mail" on a data file.

UUCP communicates by means of "protocols", the most common being "g", a method for transmission of data over telephone lines and ensuring that the data is not corrupted. There are several other protocols, none universally available, and most oriented towards communication media other than telephone voice lines (such as dialup X.25, PAD X.25, or LAN connects).

UUCP operates over fixed system-to-system links, so sending mail from one system to another often has to traverse other intermediate systems.

TCP/IP:

TCP/IP is a protocol for use over more "broadcast-like" networks. Where it is possible for any system on the network to talk "directly" to any other. TCP/IP (and it's later relative OSI 802.3) is usually used over networks built in top of Ethernet, Starlan and other LANS.

SMTP:

Or, "Simple Mail Transfer Protocol", is the communications protocol used most commonly over TCP/IP links in UNIX environments. SMTP usually operates directly between the source and destination machines, so intermediate machines don't get involved (except for gateways, see below). SMTP is run on top of TCP/IP.

SLIP:

SLIP is an implementation of TCP/IP designed for use over RS232 serial lines (ie: modems). The other difference is that some SLIP implementations have the ability to "dial the phone" to make a connection for a specific transfer, whereas LAN TCP/IP is logically continuously connected. You'd also need TCP/IP to run a SMTP mail connection.

PPP:

A successor to SLIP.

Internet:

Ah, the internet... A subject to warm the toes on a frosty night... For our purposes, suffice it to say that it is a vast loosely coupled network of 10's of thousands of machines using internet protocol mail, communicating with each other largely by TCP/IP and SMTP over relatively high speed lines. The internet grew out of a US government funded project in inter-computer communications that grew into an enormous network of systems using similar software. One of the principle characteristics of this network is that machines are addressed by domain names which identify the destination, rather than addresses that are constructed out of the route from machine-to-machine-to-machine.

UUCP Network:

The UUCP network is that set of machines that talk to each other via UUCP. Sending mail through this network requires that the sender know the network topology of UUCP links, and specify a path from one machine to the next. (There are, of course, ways around this. See the section on "do's and don'ts".)

Addresses:

An email address is a method of specifying a given person on a specific machine. There are scads of conventions, usually determined by the presence of "@"'s, "!"'s and other special characters in the name. An address usually consists of two parts: a userid/name and a machine specification.

A Domain address usually looks like:

```
userid@domain-address
```

Whereas a UUCP address usually looks like:

```
route-to-machine!userid
```

Domain Addresses:

Domains are a way of uniquely specifying a destination. Much like a postal address, a domain specifies a set of progressively more restrictive "domains" of the potential address space. It's perhaps illustrative to give an example:

clewis@ferret.marketing.fooinc.com

You read these things right to left: "com" means the commercial domain. "fooinc" is the name of an organization within the commercial domain. "Marketing" is the name of a suborganization within fooinc, and ferret gives the name of a machine (usually). Domains can have any number of levels. The top level domain (com in the above example) has many possible values. In the United States, "com", "mil", "edu", and "gov" are fairly standard. Elsewhere, the top level domain tends to be a country code, the second level tends to be a province or state, and the third an organization. (But, for example, there are many com and edu sites in Canada and other countries).

A fully-qualified-domain-name (FQDN) has a entry for each level of the domain, from individual machine to top-level domain. In many cases, an organization has implemented a organizational "gateway" at a higher level of domain, so that people from outside don't have to specify FQDN's to get to a specific person. In the above example, for instance, "fooinc.com" may be sufficient to get to anyone inside fooinc, and "ferret.marketing" may not be necessary.

Internet domains are controlled by a single organization, the NIC. In contrast, on UUCPnet, machine names are selected on a whim (though they shouldn't be - see do's and don'ts), and may clash with other machines.

Bang-Paths:

With UUCP mail, you have to specify a route to get from one machine to another. "A!B!C!userid" means go to machine A, then B, then C, then user "userid" on C.

Gateways:

There are several meanings of this term, only two are relevant here.

The first is a mechanism for getting from one network to another network that uses different protocols.

The second is a mechanism for getting from one logical (often organizational) network to another using the same protocol. Often for example, there will be a LAN in one department of an organization, and one machine in the LAN has the connection to another LAN in another department. This means that mail from one LAN to the other has to pass thru the gateway machine.

Routers:

There are several definitions, but the most important is that part of the TA that figures out how to send a message to a given machine. This often uses a database that provides routes from one machine to the other machines on the network.

Smarthost:

In many cases, your machine won't know how to get to a specific destination. You can usually set up your mail system to send mail, that it doesn't know how to deliver, to a machine that is more likely to.

RFC's:

A set of documents that include formal descriptions of mail formats used on the Internet, and are adhered to by many non-Internet systems. More specifically, in the "worldnet" of USENET, Internet and UUCP, the RFC's set the standards for mail exchange.

It should be pointed out, however, that there are some regions where the RFC's are not entirely respected. For example, British email networks uses domains, but they're specified backwards (they drive on the wrong side of the road too).

"The Maps":

A set of files describing machine-to-machine links distributed over USENET in the group comp.mail.maps. These are usually posted on a monthly schedule, and can be automatically received and transformed into a routing database that describes the "optimal" route to each machine. These are operated by the "UUCP Mapping Project". See the README posted along with the maps for more details.

Aliases:

Aliases are a mechanism by which you can specify the destination for mail on your machine. Through the use of aliases you can redirect mail to "virtual userids". For example, you should have a mail destination on your machine called "postmaster", which is aliased to send the mail to the System Administrator (ie: you probably). Aliasing often also permits you to send mail to groups of users (not necessarily on the same machine as you) pipelines of commands or to specific files.

Mailing lists:

Is similar to a USENET newsgroup. They are usually aliases pointing to groups of users, and allow mail to be sent to the whole group at once. Mailing lists are set up to carry certain subjects. The difference between a mailing list and a USENET newsgroup is that the messages are sent by mail, probably as a copy to each recipient, rather than broadcast.

Do's and Don'ts:

1) Register a domain name. Even on UUCP, where <machine>.UUCP is often

used as a kludge, it is MUCH preferred that you obtain a real domain address. If you are directly connecting to the Internet, you will get one as part of your registration with the NIC.

If you aren't connecting direct to the Internet, obtaining a registration will usually require you finding a nearby friendly internet site willing to act as a mail forwarder to you from the Internet (the site that will set up a "MX record" for you). Many sites will do this for you for free, and several of the commercial email services (eg: uunet) will do it for you for a nominal charge (without requiring you buy the rest of their services).

There are occasions where you can join what is called a "domain park". These are most often small regional groups of systems that have gotten one of their number properly registered as a domain, and provides forwarding services out to other systems. For example, in my address "ferret.ocunix.on.ca", "ocunix.on.ca" is a domain park made up of the Ottawa-Carlton UNIX User's Group, one of the other machines in the group provides a gateway between our systems and the Internet.

2) If your machine is going to "speak" UUCP to the outside world, choose a unique UUCP name. You can find out whether a name you want is taken by consulting the UUCP maps. Or by asking someone else who's using them.

3) Register your machine with the UUCP Mapping Project. Information on how to do this is included in the monthly maps postings in the file "README". This is usually only required when your machine talks UUCP to the outside world, or when other machines have to address you by your UUCP name. If you don't do this, somone else may choose the same name, and gross confusion will arise when smart routers won't be able to tell whether to send a piece of mail to you, or your doppelganger[s]. If you register with the UUCP Mapping Project, you have prior use, and people who choose the same name afterwards will be told to get a new one.

   If you're "behind" an organizational gateway, don't do this. (Your organizational gateway is the thing that needs to be registered)

3) If you're going to be setting up multiple machines, have only one or two connections to the outside world.

4) Install a mail system that understands domain addressing, even if you aren't registered. (In fact, all of the suggested configurations in this FAQ do)

5) *Never* use UUCP bang-routing if you can possibly avoid it.

6) Find a friendly neighboring SA to help. Having SA who's already operating mail in your area will help smooth over the regional "gotchas" that are bound to crop-up. And advise you on the right software to use, where to obtain it, and how to install it.

7) Do NOT use "any old" Map unpacking program. Most available map unpacking programs automatically run the shell (or shar) to unpack map articles. Since it is trivially easy to forge map articles, using this type of unpacking program can easily let very destructive trojan horse or virus programs into your machine.

   The two specific map unpackers described in this FAQ are known to be secure from such attacks. Do not run any other unpacker unless you are aware of the issues and can inspect the code for such vulnerabilities. [If you know of other "secure" map unpackers that are generally available, please let me know]

Configuration Issues:

What you need for email connectivity is determined by:

1 what networks you intend to connect to.
  The internet (hence SMTP)? UUCP sites? Others?
  Combinations?
2 what links you have or are willing to install
  Internet T1? T2? UUCP? Other? [Details on how to make your connections is beyond the scope of this FAQ, but can usually be found out from the provider (other end) of the link]
3 what user interface you want to use. This is largely an independent issue, so consult the Specific Package Reviews directly.

Recommended TA Configurations:

These configurations are based upon my own experience, and the experience of others. Careful installation of any of these configurations will result in a solid, reliable mail system that respects the appropriate "do's and don'ts". Each configuration represents a compromise of ease of installation and maintenance versus sophistication and capabilities.

One thing you should consider is what you already have on your system. You will invariably have "binmail", and will have a good chance at already having sendmail. The configurations shown below are *minimal* configurations, so you should consider whether you want to use what you already have or not. [Sendmail, in particular, you may want to avoid if you can]

Scenario 1: Only UUCP connections.

Smail 2.5. If you want to set up a routing database of your own, you will also need pathalias, and unpackmaps or uuhosts. Instead, though, you can configure smail 2.5 to smart-host most destinations to a nearby friendly site who'll do your routing for you without having to run the routing software.

Smail 2.5, as shipped, does not support mail-to-pipeline or mail-to-file aliasing. If you need these, at a minimum, you should obtain lmail. If you intend more than casual use of these features, it is recommended that you obtain deliver or procmail instead of lmail.

Even if you have sendmail already, you can integrate smail 2.5 with it to do your UUCP routing. (though, some later versions of sendmail can do routing themselves)

(See the note regarding smail 2.7 in the Package Reviews)

Scenario 2: SMTP connections (optionally, some UUCP connections too).

Generally speaking, sendmail will do this for you and you have a good chance to have it already. However, for the novice, it is recommended that smail 3 be used instead [see review of sendmail below]. Smail 3 includes all of the routing software and can do mail-to-pipeline and mail-to-file, so none of the auxiliary programs mentioned in scenario 1 are necessary.

Sendmail doesn't include UUCP routing mechanisms, so you would need pathalias and unpackmaps or uuhosts if you wish to set up a UUCP routing database. Further, most sendmails don't know how to query a pathalias database directly, so you may have to hack your own path lookup program into the sendmail.cf (smail 2.5 can be used for this purpose provided that you will have a UUCP link to the outside world)

Deliver or procmail are still quite useful in this configuration for extended alias facilities.

Scenario 3: Connections to other networks (optionally including SMTP or UUCP), or very high loading.

You are limited to sendmail or zmailer. Zmailer is easier to administer, probably more secure, and less stress on your system (and your brain ;-).

Specific packages:

Honesty requires me to point out which software packages were reviewed by their author (including me ;-). I do so by appending a "*" to the name of the author. It is worth noting, though, that each package is well known, and have all been in operation at many sites for periods of a year or more. Each of these packages do their job well, and were written by people who are well respected in this field.

binmail

binmail is a really simple UA, and is usually called "mail" on many systems. On system V, it also does dual duty as the TA. On System V systems, it's pretty awful because it doesn't know how to set up headers properly, and often doesn't even know what a "Subject:" line is, and there's no way to do any kind of aliases.

On BSD, mail invokes sendmail to do the TA function. On System V you really do want to replace binmail's TA functionality with something else. However, you should not replace it in it's "mail" (UA) functionality, because many system-level administration mechanisms will break. Any new UA should be installed as a different name than "mail".

rmail

binmail's TA functionality is implemented by linking mail to rmail. It's rmail that you'd want to replace with smail 2.5 etc.

Mail

The original BSD UA. It can support local profiles, aliases, folders, header previewing, out-going mail recording and all sorts of good stuff. An "okay" UA. Available from BSD "freed-sources" archives.

mailx

AT&T's answer to BSD "Mail". They are very similar. Though, some versions (in particular the 3b1 one) should be avoided because they're so buggy. Not available in source form.

mush: author Dan Heller <argv@sun.com, argv@garp.mit.edu>

The "Mail User's Shell" is quite similar to Mail and mailx in basic functionality, but has been taken FAR beyond either. It has, for example, the ability to select articles and pipe them through programs, reconfigure display format etc. A very good UA - if you like the ideas in Mail or mailx, you'll really love mush. Mush also has a SunTool interface. Latest version is 7.2.3. Last posted in comp.sources.misc volume 18 (with subsequent patches).

elm [I have no availability/author/version info. This review is lousy. Help!]

Elm is a full screen UA with pretty sophisticated functionality.

Its afficionados are legion and committed. It's a particularly good UA for novices.

uumail:

Uumail is a very old and obsolete precursor to smail 2.5. Included here only because I know that uumail sites still exist. You should not install uumail in new configurations, and existing uumail sites should convert to something more modern.

smail 2.5: author The UUCP Mapping Project

Smail 2.5 is a simple, hard-coded rules mailer for use on UUCP networks. It understands RFC compliant headers, will generate RFC compliant Internet-style headers, can use domains, aliases, a pathalias UUCP routing database, and is very simple to install. For full functionality, you will also want pathalias and a map unpacker. The one thing it cannot do by itself is mail-to-pipe and mail-to-file aliasing. For that, you need Zeef's lmail, deliver or procmail.

Smail 2.5 has a few bugs that are (usually) pretty rarely seen operation. There have been a number of patches posted for it, but it is recommended that you do not apply them - some were ill-conceived, buggy in their own right, or conflicting with others. The only patches that I feel safe in recommending is Chip Salzenberg's patches for use with Xenix MICNET - which are unnecessary unless you are in the unfortunate position of having to actually *use* MICNET. If you're concerned about the bugs, I suggest you wait for smail 2.7.

Smail 2.5 can also be used with sendmail as a UUCP router.

smail 2.5 was posted in comp.sources.unix in 1987, volume 18.

smail 2.7 custodian: Greg Woods <woods@eci386.uucp>

Smail 2.5 was originally a dead end. There was a smail 2.6 used internally in some places (no, don't ask for it). Greg Woods is doing a major cleanup and revision of smail 2.[5-6] (with permission of the authors) and will be releasing it (I hope) soon. Smail 2.7 will be, more-or-less, a version of smail 2.5 with all bugs eliminated, and lmail's functionality rolled in. It still won't include pathalias or a map unpacker. (right Greg?)

lmail author: Jon Zeef.

When you install smail 2.5, you link the original /bin/mail (binmail above) to /bin/lmail to perform the task of actually delivering the mail to the user's mailbox.

Since smail 2.5 was not capable of doing mail-to-pipe and mail-to-file aliasing, Jon Zeef wrote a replacement lmail that implemented these (along with user mailbox delivery).

Jon's program is okay for casual use, but has some pretty serious bugs. Fixed versions are available, but you're probably better off waiting for smail 2.7, or installing deliver or procmail.

smail 3: Author Ronald S. Karr* <tron@veritas.com> and Landon Curt Noll.

Smail3.1 is a domain-capable mail router and delivery program that works in the UUCP zone and on the Internet and that is capable of gatewaying between the two. It was written primarily by me (Ronald S. Karr) and Landon Curt Noll, with the blessings of the original Smail1 and Smail2 authors.

Smail3 supports SMTP, UUCP mail, alias files, .forward files, mailing list directories, pathalias files, /etc/hosts files, the domain name system, and can also query uucp for neighboring sites, automatically. It also supports use of encapsulated SMTP commands for delivery over UUCP connections, which allows batching of multiple messages into a single UUCP transaction, and allows many addresses to be passed with a single message transfer, which can greatly decrease the traffic generated for large mailing lists. It is also very simple to configure with a reasonable certainty of correctness.

Smail3 includes pathalias and a reliable map unpacker.

Rather than using configuration files to resolve addresses based on their syntax, ala sendmail, Smail3 uses a database metaphore for resolving addresses based on their contents. The set of methods that Smail3 uses for resolving local addresses and hosts is configurable and extensible. Smail3's methods for parsing addresses are not configurable. It is the opinion of the authors that addressing on the Internet and in the UUCP zone has become sufficiently standardized that attempts to allow configurability in this area are now a hindrance to the correct working of the network.

Current releases of Smail3 have some problems when operating on the Internet or on local ethernet networks. Some of the problems arrise when when the level of traffic gets too high, and result from the fact that Smail3 does not currently limit its usage of CPU resources. Additionally, Smail3 is not yet sufficiently intelligent in its management of outgoing SMTP connections. As a result, Smail3 does not make effective use of single connections, and cannot use a history of connection failures to alter its behavior. These problems are now being addressed.

Questions related to Smail3 are usually discussed in comp.mail.misc. There are also two discussion mailing lists. To join the mailing lists, send mail to:

smail3-users-request@cs.athabascau.ca
smail3-wizards-request@cs.athabascau.ca

The current release of Smail3 can be found on uunet, in the directory /usr/spool/ftp/mail. The current release file is smail-3.1.23.tar.Z.

sendmail: author er, ah, h'm... Good question

sendmail is the granddaddy of all intelligent TA's. It can do just about anything. It's main problem is that it can do just about anything. Modification of sendmail's configuration tables (which is usually necessary) is NOT for novices. The language of the sendmail.cf is cryptic, but that isn't really the problem (and this problem can be solved by using "EASE", a sendmail.cf writing language). The problem is that, without a PhD in RFC reading, it's extremely difficult to know when the rules you are implementing are the right thing - most people's sendmail configurations do slightly buggy things - eg: some major mail gateways munge From: headers in illegal ways.

Worse, every vendor's versions of sendmail is different, and many of their sendmail.cf's don't work at all (HPUX is one example of where the sendmail.cf is actually pretty sane. HP is to be congratulated. On the other hand, some vendors, who shall remain nameless can't even get their sendmail to speak SMTP on a LAN).

There are several versions of sendmail extent in the world. There is no "latest" version. If you MUST use sendmail, it is strongly recommended that you obtain a version of sendmail with the "IDA patches" installed. The IDA packages make sendmail a little more likely to do the right things with mail coming from, or ultimately going to, UUCP sites.

ZMailer: Author Rayan Zachariassen* <rayan@cs.toronto.edu>

ZMailer is intended for gateways or mail servers or other large site environments that have extreme demands on the abilities of the mailer.

Code and Design features:

+ Strong limits on host impact.
+ Secure design (and hopefully implementation).
+ Natural fit for client/server environments.
+ Extremely customizable configuration mechanism.
+ Flexible database interface with support for: sorted files, unsorted files, dbm, ndbm, gdbm, nis (yellow pages), dns (BIND resolver), /etc/hosts file, and in-core data.
+ Efficient message queue management.
+ Fast binary-transparent SMTP server and client.

+ Low-technology implementation.

Default configuration file features:

+ Default configuration will work for most sites.
+ Network protocol support for: smtp, uucp, bitnet, mail to news.
+ An easy way of overriding any external routing information.
+ Automatic handling of mailing lists.

It is available by anonymous FTP from ftp.cs.toronto.edu:/pub/zmailer.tar.Z
or ftp.cs.toronto.edu:/distrib/zmailer.tar.Z.

mmdf [help!]

deliver: Author Chip Salzenberg* <chip@tct.com>

Deliver allows any user to write a shell script that processes all incoming mail messages for that user. The system administrator may also install scripts that process all messages by installing it as the Local Delivery Agent (lmail replacement).

The output of a script is a list of mail addresses, files and programs that should receive the message. It has access to each message as it is processed, so the action can be content dependent. The script may also generate automatic replies, like the "vacation" program, or pass along a modified version of the original message.

Deliver can be used to construct mail-based services (e.g. automatic mailing list maintenance). It can also be used to filter mail automatically in prearranged ways (e.g. encryption and decryption, tossing junk mail, or vacation notices).

Deliver was last posted in comp.sources.reviewed. The current version is 2.08.

procmail: Author Stephen R. van den Berg*
<berg@messua.informatik.rwth-aachen.de>

Can be used to create mail-servers, mailing lists, sort your incoming mail into separate folders/files (real convenient when subscribing to one or more mailing lists or for prioritising your mail), preprocess your mail, start any programs upon mail arrival (e.g. to generate different chimes on your workstation for different types of mail) or selectively forward certain incoming mail automatically to someone.

The accompanying formail program enables you to generate autoreplies, split up digests/mailboxes into the original messages, do some very simple header-munging, or force mail into mail-format (with leading From line).

Feature summary for procmail:
+ It's small

```
+ Very easy to install (rated PG6 :-)
+ Simple to maintain and configure because
  all you need is actually only ONE executable (procmail)
  and ONE configuration file (.procmailrc)
+ Is event driven (i.e. gets invoked automagically when mail arrives)
+ Does not use *any* temporary files
+ Uses standard egrep regular expressions
+ Allows for very-easy-to-use yes-no decisions on where the mail
  should go
+ Filters, delivers and forwards mail *reliably*
+ Provides a reliable hook (you might even say anchor :-) for any
  programs or shell scripts you may wish to start upon mail arrival
+ Performs heroically under even the worst conditions
  (file system full, out of swap space, process table full,
  file table full, missing support files, unavailable executables,
  denied permissions) and tries to deliver the mail somehow anyway
+ Absolutely undeliverable mail (after trying every trick in the book)
  will bounce back to the sender (or not, your choice)
+ Is one of the few mailers to perform reliable mailbox locking across
  NFS as well (DON'T use NFS mounted mailboxes WITHOUT installing
  procmail, you may use valuable mail one day)
+ Supports four mailfolder standards: single file folders (standard
  and nonstandard VNIX format), directory folders that contain one file
  per message, or the similar MH directory folders (numbered files)
+ Variable assignment and substitution is an extremely complete subset
  of the standard /bin/sh syntax
+ Provides a mail log file, which logs all mail arrival, shows
  in summary whence it came from, what it was about, where it went
  (what folder) and how long (in bytes) it was
+ Uses this log file to display a wide range of diagnostic and error
  messages (if something went wrong)
+ Processed mail can contain arbitrary 8-bit characters (including
  '\0'); i.e. binary mailings can be processed if the rest of the
  mailing system knew how to handle them too
+ It has a man page (boy, does *it* have a man page)
+ procmail can be used as a local delivery agent (a completely
  integrated substitute for /bin/mail)
+ It runs on virtually all (old and future) operating systems which
  names start with a 'U' or end in an 'X' :-) (i.e. extremely portable
  code; POSIX,ANSI C and K&R conforming)
+ Works with (among others?) sendmail, smail and MMDF

   Feature summary for formail:
+ Can generate auto-reply headers
+ Can force mail into mailbox format (so that you can process it with
  standard mail programs)
+ Can split up mailboxes into the individual messages
+ Can split up digests into the individual messages
+ Can split up saved articles into the individual articles
+ Can do simple header munging
```

```
    Feature summary for lockfile:
+ Provides NFS-secure lockfiles to shell script programmers

    A recent version can be picked up at various comp.sources.misc archives.
    The latest version (v2.40) will be posted in comp.sources.misc shortly,
    and is already available directly from the ftp-archive at:

amaru.informatik.rwth-aachen.de (137.226.112.31)

as compressed tar file: pub/unix/procmail.tar.Z
or in compressed shar format: pub/unix/procmail.0?.Z

pathalias: Author Peter Honeyman

    Pathalias reads the UUCP Map Project maps (they need to be extracted
    from the postings first) and constructs a database containing the
    minimum cost route to any machine in the maps.  This database can
    then be used with any mailer that knows how to search the database
    (eg: smail 2.5, Zmailer?, and some versions of sendmail.  Smail 3
    comes bundled with pathalias).

    There were previous versions of this program.  You must use
    pathalias version 10 (latest version), because some map format
    changes have been made and only pathalias 10 can parse them.

    There were other route-generating programs, but all (as far as
    I know) are very obsolete, and none run as fast as pathalias
    (still, which can be rather hard on machines with smallish virtual
    memory or RAM capacities).

uuhosts: Author John Quarterman (?)

    The "defacto" standard UUCP Map Project map unpacker.  Includes
    a program to arbitrarily view individual map entries.  Uuhosts
    implements trojan horse/virus security by running under
    a "chroot()" system call.  Uuhosts does not appear to be
    actively maintained, and the last versions that I have inspected
    were unable to easily compress the maps (a full set of maps
    is >6000 blocks), had no provision for automatically
    running pathalias, and will not work with the newest version
    of cnews.  Further, uuhost's header checking is so picky
    that the slightest change in the map format will cause
    uuhosts to reject map updates.

    Use of uuhosts now will require some minor hacking.

    The last edition, "uuhost4" (version 1.69) appears to have
    been posted in comp.sources.unix in volume 3, 1986.

unpackmaps: Author Chris Lewis* <clewis@ferret.ocunix.on.ca>
```

```
    Unpackmaps has a superset of the functionality of uuhosts.
    It obtains its security by doing the map unpacking with
    a specialized awk script that knows the map article format
    rather than invoking a shar.  Compression and pathalias
    invocation is automatic, correctly takes into account
    the change date of local configuration files, and will
    work with the latest Cnews.

    The last version of uuhosts, version 3.2, was posted in
    comp.sources.misc in Aug 1991 (volume 22, issue 69).
    (do a g/NEWSPOOL/s//NEWSSPOOL/ on unpackmaps to fix a
    botch with cnews).  Version 3.3 will come out as soon
    as this FAQ is finalized ;-)
--
Chris Lewis; clewis@ferret.ocunix.on.ca; Phone: Canada 613 832-0541
Psroff 3.0 info: psroff-request@ferret.ocunix.on.ca
Moderator of the ferret list: ferret-request@ferret.ocunix.on.ca
**** NEVER let something mechanical know you're in a hurry. ****
```