The Elm Alias System Users Guide

What aliases are and how to use them
in the **Elm** *mail system*

Dave Taylor

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto CA
94304

email: taylor@hplabs.HP.COM or hplabs!taylor

>>> Elm is now in the public trust. Bug reports, comments, etc. to: <<<

Syd Weinstein
Datacomp Systems, Inc.
3837 Byron Road
Huntingdon Valley, PA 19006-2320

email: elm@DSI.COM or dsinc!elm

The Elm Alias System Users Guide

(Version 2.3)

Dave Taylor

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto CA
94304

email: taylor@hplabs.HP.COM or hplabs!taylor

>>> Elm is now in the public trust. Bug reports, comments, etc. to: <<<

Syd Weinstein
Datacomp Systems, Inc.
3837 Byron Road
Huntingdon Valley, PA 19006-2320

email: elm@DSI.COM or dsinc!elm

May 1, 1990

This document is intended as a supplement to the *Elm Users Guide* and is only of interest to those users desiring more knowledge about how aliases work and how to create strange and exciting aliases for their systems (okay, so maybe it's not *that* exciting!)

This document is broken up into the following sections; user aliases, group aliases, system aliases, editing and installing new aliases, the host routing database, the domain routing database, general warnings and other chitchat.

1. User Aliases

The most simple type of aliases in the **Elm** system are individual user aliases. These are made up of three parts;

aliasname list = username = address [1]

1. Notice that the alias separator has moved from being a colon character (':') to the equals sign ('='). This is due to the requirements of the X.400 addressing scheme.

Where the *aliasname list* is either a single aliasname[2] or a list of aliasnames separated by commas. Aliasnames are case insensitive and will be converted to all lower case by the alias system.

Username is used to indicate the full 'real name' of the user. For example, if you had an alias for 'taylor' to get to me, the *username* field would contain 'Dave Taylor' or perhaps 'Dave Taylor at HP' or some other permutation. **Elm** uses this information to add the users full name to the addresses of outbound mail to get more readable addresses.

Address is either the users full electronic mail address or, if the machine routing database is installed, the minimum address needed to specify the destination. For example, say our routing database contained information on how to get to machine 'hp-sdd' and I wanted to have an address for my friend Ken there — I could have his address specified as simply 'ken@hp-sdd' (or alternatively 'hp-sdd!ken' since the two are functionally equivalent).

Let's get on to some examples, shall we?

Consider this excerpt from my own *.elm/aliases.text* file;

```
# Mail aliases for Dave Taylor
# Friends from UCSD
addie          = Addie Silva    = addie@hp-sdd.SDD.HP.COM
frank,minjarez = Frank Minjarez = Minjarez.Verify@dockmaster.arpa
pizzini        = Ken Pizzini    = hplabs!ucbvax!ucdavis!pai!ken
```

Note that the alias for Frank Minjarez has two *aliasnames* associated with it, `frank` and `minjarez`. Also notice that the first and second aliases use the ARPA Internet style naming convention of *user@machine* whereas the third uses the UUCP notation of *machine!user*.

For the most part, the notational format is transparent within the alias file — the only time it *does* make a difference is if you have to specify more than the machine that the user is receiving mail on (as in the third example above).

Suffice to say that if you must specify any sort of uucp route that you should use the uucp notation as much as possible to ensure that the system expands the correct machine name. Similarly, if you're bouncing mail through different ARPA Internet sites (to get to a foreign system, as in the example below) you should use the notation that system expects:

```
listserv%byuadmin.BITNET@rugters.edu
```

2. Group Aliases

After the confusion of user aliases, group aliases are even more fun! For the most part the notation is very similar;

aliasname list = *groupname* = *list of people*

Where *aliasname list* and *groupname* are exactly equivalent to the corresponding fields in user aliases.

The interesting part is the *list of people* field; this field is actually in the same notation as the aliasname list, so it's really quite easy to create. It's best to illustrate by example:

```
friends, mypals = The Gang of Six = joe, larry, mary, joanna,
                                    nancy, michael
```

2. aliasnames can be any combination of letters, numbers, dashes ('-'), periods ('.'), or underscores ('_'). Letter case is not significant. That is 'FRED' and 'Fred' are identical.

(Notice that you can continue onto as many lines as you'd like so long as each additional line start with either a SPACE or a TAB character)

The major limitation with group aliases is that each of the people in the list must be a *previously defined alias* or a valid mail address on the current machine.

What does this mean? That the following excerpt from an alias file:

```
hawaii = The Hawaiian Twins = joe@cs.rit.edu, maoa
maoa   = Maoa Lichtenski Jr = maoa@Hawaii.cs.uh.edu
```

will fail for two reasons — not only does the group *list of people* contain a complex address, but it also contains an aliasname that is defined *further on* in the *.elm/aliases.text* file.

The correct way to have the previous aliases in the file are to have them organized like:

```
joe    = Joe Lichtenski     = joe@cs.rit.edu
maoa   = Maoa Lichtenski Jr = maoa@Hawaii.cs.uh.edu
hawaii = The Hawaiian Twins = joe, maoa
```

which will then work correctly.

Fortunately, while this seems fairly picky, when you run *newalias* to install the new aliases, you'll get relevant and meaningful error messages to help you fix your aliases up correctly.

3. System Aliases

System aliases are functionally equivalent to the individual **Elm** alias lists each **Elm** user has (both user aliases and group aliases) but are *read only* for everyone but the **Elm** administrator. The format of the file is identical to the users file, and the only difference is that this file is expected to be located in the directory that contains the *system_hash_file* and *system_data_file* files (see the *Elm Configuration Guide* for more details on these variables).

Simply create the system alias file in the specified directory as you would a normal alias file, and install it the same way (see the following section for more details on that).

Voila!!

4. Editing and Installing New Aliases

To install new aliases, you need merely to create, or modify, the file *.elm/aliases.text* until you're satisfied with it and it meets the requirements discussed above. You can then try to install it with the command;

$ **newalias**

which will either report back the number of aliases installed or any errors encountered trying to parse and store the given alias list.

Note that blank lines are no problem and that comments are not only allowed but actually encouraged, and must have '**#**' as the first character of each comment line.

Finally, if you find that you're hitting the ''Too many aliases'' error, then you'll need to reconfigure the entire **Elm** system (again, see *The Elm Configuration Guide*).

5. The Hostname Routing Database

Floating about on the various networks is a rather nifty program by a number of people, including Peter Honeyman and Steve Bellovin, called *pathalias*. What this software does is take the Usenet articles in the group ‘‘comp.mail.maps’’ and change them into a file of the form;

hostname <tab> *address*

which then must be sorted alphabetically and stored in the file pointed to by *pathfile* for **Elm** (and other programs) to use.

If you don’t have the program, or don’t want to use it, you can simulate this file by listing machines in the same format. The exact format expected is;

hostname<tab>*machine-address*

where *hostname* is a limited identifier (no special characters) and machine-address MUST contain the sequence ‘%s’ (and consequently any other percent signs that appear in the address must be paired) so that the call in the program ‘‘sprintf(buffer, machine-address, username)’’ will generate a valid return address. This file must be sorted alphabetically for Elm to use it properly.

By way of example, here are a few entries from my own file;

```
HPL        %s@hplabs
PARC       %s%%Xerox.PA.COM@hplabs
amc-hq     %s%%AMC-HQ.ARPA@hplabs
imsss      %s%%IMSSS%%AI.STANFORD.EDU@hplabs
infopro    hplabs!infopro!%s
interleaf  sun!interleaf!%s
```

As you can see, the addresses can get pretty complicated. In fact it’s due purely to the complexity of addresses that this file is so useful.

Some brief examples:

```
$ elm joe@HPL
To: joe@hplabs
$ elm michelle@imsss
To: michelle%IMSSS%AI.STANFORD.EDU@hplabs
$ elm fiedler@infopro
To: hplabs!infopro!fiedler
```

If you’d like further information on the *pathalias* program, try keeping track of the entries in the netnews group *comp.sources.unix* — it’s posted about once a year or so...

6. The Domain Routing Database

Another interesting feature of the **Elm** system is the shorthand domain routing database. This is the same database (in the same strange format) as used by the publically available *uumail* program.

In a nutshell, the file contains information of the form;

domain, path-to-gateway, rewrite-template

The *domain* field must begin with a leading ‘.’ and is usually ordered in the same way as the standard domain entries (that is, ‘‘.HP.COM’’ not ‘‘.COM.HP’’).

Path-to-gateway is routing information on how to get to the particular gateway that this domain expects, and always is a machine/host name (to be found in the pathalias database, see the previous section) preceded by a ‘>’ character.

Rewrite-template is the most interesting of the three, and is akin to a printf string for C. The changes are that instead of ‘%s’ ‘%d’ and so on, the actual ‘‘percent’’ values represent various parts of the address, namely;

Symbol	**Represents**
%U	The username in the To: address
%N	The remote machine name
%D	%N + domain information
%R	path to %N from pathalias
%P	*path-to-gateway* entry
%%	The ‘%’ character

with this very un-intuitive setup, let’s look at a few entries from the domains database and then see how they work:

```
.EUR.UUCP, , , %R!%U
.ATT.COM, >att, , %P!%D!%U
.HP.COM, , , %R!%U
.UUCP, , , %R!%U
.COM, >hplabs, , %P!%U@%D
.CSNET, >hplabs, , %P!18:12:00%%D@CSNET-RELAY.ARPA
.EDU, >hplabs, , %P!%U@%D
```

(Note the presence of a third field that is always null. This is for compatibility with the *uumail* program. Also notice that there is a space after each and every comma, even if that field has an entry.)

To see how it all works, let’s suppose that we want to send a message to ‘jad@cs.purdue.edu’. This would break down into the following fields:

%U = *jad*
%N = *cs*
%D = *cs.purdue.edu*

When the **Elm** program matches the *.edu* entry

```
.EDU, >hplabs, , %P!%U@%D
```

the other field instantiated would be:

%P = *path to hplabs*
template = *%P!%U@%D*

If, say, our path to *hplabs* was *hpcnoe!hplabs*, then the fully expanded address would be:

```
hpcnoe!hplabs!jad@cs.purdue.edu
```

and so on.

What does this mean to the average user? It means that you can for the most part send mail to people on different networks by simply using their full address, including any domain information, so that mail to addresses like ‘‘Jack@AI.MIT.EDU’’ will work, a message to ‘‘SueAnn@BBN.MAILNET’’ will work and so on!

7. Other Stuff not Covered Yet

Probably the biggest question you have in your mind right now is ‘‘But how the heck does this relate to my existing *Berkeley Mail* aliases and the lower-level *sendmail* alias system??’’ Well, rest assured that if you *really* want to have your aliases down in the transport you can. No problem. All you’ll need to do is to turn off the address validation routine in **Elm**.

And for those ex-*Berkeley Mail* fans, you can translate your aliases into the format that **Elm** wants by running them through the *awk* script listed in the appendix.

Finally, if you have any problems or questions, try looking in the *newalias* manual entry, or dropping me a line at the ‘‘usual’’ email address.

Appendix One
An AWK Script for Translating Aliases from
a *Berkeley Mail* ‘‘.mailrc’’ File to an *Elm* ‘‘.elm/aliases.text’’
File

```
BEGIN { print "# Elm .elm/aliases.text file, from a .mailrc file..."
      print ""
      }
next_line == 1 {
      next_line = 0;
      group = ""
      for (i = 1; i <= NF; i++) {
        if (i == NF && $i == "\\") sep = ""
        else                       sep = ", "
        if ($i == "\\") {
          group = sprintf("%s,", group)
          next_line = 1;
        }
        else if (length(group) > 0)
          group = sprintf("%s%s%s", group, sep, $i);
        else
          group = $i;
        }
        print "\t" group
      }
$1 ~ /[Aa]lias | [Gg]roup/ {
      if ( NF == 3)
        print $2 " = user alias = " $3;
      else {
        group = ""
        for (i = 3; i <= NF; i++) {
          if (i == NF && $i == "\\") sep = ""
          else        sep = ", "
          if ($i == "\\") {
            group = sprintf("%s,", group)
            next_line = 1;
          }
          else if (length(group) > 0)
            group = sprintf("%s%s%s", group, sep, $i);
          else
            group = $i;
          }
          print $2 " = group alias = " group;
        }
      }
```

Note: this script is contained in the release under the name ‘‘mailrc.awk’’ in the utilities directory ‘‘utils’’.