Chapter 3

Distribution List Maintenance

3.1 File based Distribution Lists

PP provides an interactive tool for maintaining distribution lists, *mlist*. This tool allows you, the user, to view, modify and generally manage the members of a given list. What you are allowed to do to a list depends on the access rights set on that list.

The access rights for a list are set when the one of the managers of that list first uses *mlist* to manage the list. There are four modes of access rights:

free: Anyone can add or remove members of the list.

public: Anyone can add themselves to or remove themselves from the list.

private: Only managers of the list can add or remove members;

secret: Only managers of the list can view or alter the members of a list.

If you are a manager of a list, the access rights are bypassed and you are given *free* access to the list.

3.1.1 What you can and cannot do with *mlist*

For all modes of access except *free*, the maintenance of the list is done via a series of questions. For example:

```
You (pac) are not in this list
Do you wish to be added ? [y/n]
```

With *free* access, the maintenance is done via a menu of options.

add: A list of possible mail addresses is expected with this option (either following the add command or on the next line). *mlist* checks that each address is a valid mail address and that the normalised address does not already exist in the list. Addresses that pass both these tests are added to the list.

remove: A regular expression is expected with this option. Any address that matches this expression is a candidate for removal from the list. *mlist* asks the user whether each candidate should be removed.

find: As with the *remove* option, a regular expression is expected. Any address that matches the expressions is printed out.

verify: This option goes through the list checking that each member is a valid mail address.

As well as these maintenance options, there are a few inspection options

print: This option prints out all the information concerning the list. This information consists of all the members of the list, the list moderator's mail address, and the user ids of the moderators of the list.

list: This option displays all the lists known to the local PP system. Each list also has a short description of its purpose.

help: This option displays all the options that are available to you.

If the list you require does not exist, there is one further option, `create`. This option prompts you for all the details of the list you require and posts them to the postmaster. If the postmaster decides to grant you this list, he or she will then do the appropriate administration to create the list. Once the postmaster has done this, you can use *mlist* to manipulate the list.

If you have done this and the list you requested now appears in the output of the `list` option but *mlist* still thinks it does not know about that list, then it is possible that the postmaster overlooked creating an empty file to contain the list when he or she was adding the list to PP's tables.

3.1.2 How *mlist* can run

mlist takes each list specified on the command line in turn, and enables you to modify it. If no lists are specified on the command line, *mlist* enters a loop. Within this loop, you choose the lists you wish to modify. When you have finished modifying the required lists, you can choose to `quit` the loop and exit *mlist*.

mlist can take several command line arguments:

-user 'username' This specifies the username one wants to be known as when altering lists. Only privileged users can use this flag.

-uid 'userid' This specifies the user Id one wants to be known as when altering lists. Only privileged users can use this flag.

-order 'domain order' This specifies the preferred domain ordering to use when parsing addresses. This can be specified as:

- **usa** usa domain ordering ONLY.
- **uk** uk domain ordering ONLY.
- **usapref** usa domain ordering preferred.

ukpref uk domain ordering preferred.

The default is usa domain ordering only.

3.1.3 *malias*.

The *malias* tool is a passive version of *mlist*. Using the *malias* tool, you can only view the members of specified lists. You are unable to modify, in any way, the contents of a list.

3.2 X.500 Directory based distribution lists

Directory lists use the OSI directory to store the list of recipients. This has the advantage that the list can be easily viewed using any Directory User Agent (DUA), making the management task easier. Some knowledge of the QUIPU Directory service is assumed.

3.2.1 The Basics

The manager of the system must set up a directory list in the user table for PP to be able to expand them. Once this is done, the maintenance of the list can be delegated to anyone with the appropriate permissions. To set up a local directory based list called `raboof`, an entry in the *users* table is needed. This should look something like

```
raboof:dirlist lancaster.xtel.co.uk
```

Each mail list needs a manager. Typically this is the mail list name with `-request` appended, `raboof-request` in this case. So the PP *aliases* table may contain something like:

```
raboof-request:alias c.robbins
```

When mail for `raboof` comes in, the directory list channel is run The first action is to convert the `raboof` name into a Directory Name. First the ISODE *isoaliases* in the *ETCDIR* is consulted. If there is an entry of the form

```
raboof  "c=GB@o=Site@cn=The Raboof List"
```

Then the given directory name is used. If no such entry is found, then a one level directory *search* operation is used. The base of the search is defined by the `local_DIT` variable in the *dsaptailor* file in the *ETCDIR* directory. The search looks for entries belonging to the objectclass `ppDistributionList` with a `commonName` attribute of `raboof`. [1]

Finally, the entry is read and the mail is delivered to the indicated recipients.

[1] `raboof` does not have to be the RDN of the entry, just an attribute. However you should ensure that the above search will only match one entry. If in doubt use an alias.

3.2.2 The DIT Entries for the list

The main entry, as described above must belong to the object class `ppDistributionList`. This has the following mandatory attributes:

`mhsDLMembers`: The ORName of members of the list. Section 3.2.3 discusses the format of ORNames in detail.

`owner`: The DN of the owner of the list. Typically, this will be a role entry for the `-request` of the list. The object class `ppRole` is useful for this as it allows you to specify an `mhsORAddress` for the role. A typical entry might be:

```
cn= The Raboof List Manager
cn= raboof-request
roleOccupant= c=GB@o=X-Tel Services Ltd@cn=Colin Robbins
mhsORAddresses= /S=Raboof-request/O=XTEL/PRMD=X-Tel Services/ADMD= /C=GB/
```

`cn`: The name of the list.

`mhsDLSubmitPermissions`: This defines who can send to the list. The default is a special token "`ALL`" which permits anybody to send mail to the list. However it is possible to restrict the use of the list. The syntax is:

```
<permission> ::= <group> | <individual> | \
                 <member> | <pattern>
<group> ::= 'GROUP #' <DN>
```

A member of the X.500 group `DN` can send to the list.

```
<individual> ::= 'INDIVIDUAL #' <ORName>
```

The individual defined by `ORName` may send to the list.

```
<member> ::= 'MEMBER #' <ORName>
```

A Member of the named list may send. This only works if the named list is also directory based.

```
<pattern> ::= 'PATTERN #' <ORName>
```

Anybody whose ORName Matches the <ORName> pattern can send. E.g., if <ORName> was "c=GB", then only people in "GB" could send to the list.

`mhsORAddresses`: The mail address of the list, in this case

```
/S=raboof/O=XTEL/PRMD=X-Tel Ltd/ADMD= /C=GB/
```

The optional attributes are listed below.

`dl-policy`: The list delivery policy. It is generally best to leave this to the default value of: further list expansion allowed; use the original parameter for the conversion prohibited header; use low priority.

To change the default the syntax is

```
<policy> ::= <expansion> $ <conversion> $ <priority>
<expansion> ::= 'TRUE' | 'FALSE'
```

If `TRUE` sub lists are also expanded.

```
<conversion> ::= 'ORIGINAL' | 'TRUE' | 'FALSE'
```

Is message content conversion allowed. `ORIGINAL` implies use the parameter from the original message.

```
<priority> ::= 'ORIGINAL' | 'HIGH' | 'LOW' | 'NORMAL'
```

The priority to use when delivering the message. `ORIGINAL` implies use the parameter from the original message.

`description`: A textual description of the list.

The following attributes may be present but are not used by the directory list channel.

`o`: The associated organisation (if any)

`ou`: The associated organisationalUnit (if any).

`seeAlso`: A reference to another directory entry.

`mhsDeliverableContentTypes`: Acceptable content types.

`mhsdeliverableEits`: Acceptable encoded information types.

`mhsPreferredDeliveryMethods`: A list of preferred delivery methods.

3.2.3 ORNames

The most important part of directory lists is the originator-recipient name (ORName) of members of the list. An ORName consists of an optional directory name (DN), a dollar ("$") separator and an originator-recipient address (ORAddress).

ORAddresses can be specified as text encoded X.400 addresses

```
/S=Robbins/O=XTEL/PRMD=X-Tel Services/ADMD= /C=GB/
```

or as rfc822 mail addresses

```
C.Robbins@xtel.co.uk
```

In the later case, this will be converted into a X.400 style address. This conversion is useful as a short cut in giving all users with a `rfc822mailbox` attribute in the directory a `mhsORAddress` attribute.

To specify an ORName as an ORAddress only (with no DN component) you need to precede the ORAddress with a "$" symbol [2]. So, the above ORAddresses would become

[2] If the programs were smart enough, this could probably be omitted in most cases, but it has the advantage of making the user think, and hopefully use a DN or UFN instead

```
$ /S=Robbins/O=XTEL/PRMD=X-Tel Ltd/ADMD= /C=GB/
```

and

```
$ C.Robbins@xtel.co.uk
```

as ORNames. It the "$" is omitted, the interfaces will try to reverse map the mail address onto a DN — If the user does have a directory entry, it often succeeds!

It is much preferable to use a DN instead of a mail address. The DN component can be specified in one of two ways, either as a fully qualified, QUIPU style DN:

```
@c=GB@o=X-Tel Services Ltd@cn=Colin Robbins
```

or as a UFN

```
Robbins, xtel, GB
```

Neither of these examples give an ORAddress. The interfaces will determine this for themselves using the directory. If you do want to specify an address, then you can use the "$" separator:

```
Robbins, xtel, GB $ C.Robbins@xtel.co.uk
```

3.2.4 The DL tool

The program DL can be used to create and maintain the directory entries for list management, and is a very useful companion to the PP enhanced version of DISH.

To invoke DL use the *dl* command. It will read your *.quipurc* to obtain your directory username and password. The username can be changed with a "-u" flag, you will be prompted for a password. As with DISH you can request connection to a different DSA with the "-c" flag.

DL attempts to interpret what you can do to the list in terms of your access rights, based on the DN you bound with. If you are the DSA manager, you can use the "-m" flag to tell DL you have full access rights.

On entry to DL, typing a "?" will offer some limited help.

When you enter DL you will be placed at the default position in the DIT, as defined by the variable `local_DIT` in the *dsaptailor* file. You can alter this using the "m" command, and supplying a UFN:

```
> m
Currently at 'X-Tel Services Ltd'
Enter new location: cs, ucl, gb
Searching...
Found exact match(es) for 'GB'
Found good match(es) for 'ucl'
Found good match(es) for 'cs'
Moved!
>
```

You can find out which lists exist at a given level in the DIT with the "`l`" option:

```
> l
Found the following lists:
  GB Directory Pilot
  JNT PP
  JNT Quipu
  X.500 Pilot Support
  xtel
>
```

The "`w`" option will tell you which of those lists you are on.

3.2.5 Creating a List

A list can be created in one of two ways. Either an existing PP file based list can be upgraded, or a list created from scratch.

If you are bound to the directory as a "normal" user you will not have access rights to modify the directory. So both for upgrading and list creation, the DL program will ask the user a few simple questions and then send a mail message to the postmaster.

If you are bound as the DSA manager (using the "`-m`" start up flags to DL), the DL program will attempt to make the necessary changes to the directory for you.

Creating

To create a list use the "`c`" option. in the DL program. You will be prompted for the name of the list, the owner of the list (UFN), and textual description.

DL will create `listname-request` and `listname` entries in the directory.

Finally DL advises you to alter the PP tables to add the new list.

Upgrading

Upgrading is achieved using the "`u`" option in the DL program. You will be asked for the name of the PP file based list.

DL will create `listname-request` and `listname` entries in the directory. The members of the list will be read from the file based list, and added into the directory list. As many as possible will be reversed mapped into directory names.

Finally DL tells advises you to alter the PP tables to change from a `list` to a `dirlist`.

3.2.6 Managing a List

Once created the DL program can be used to manage a list. To manage the `raboof` list you can either invoke DL with the list name as an argument

```
% dl raboof
raboof>
```

or run DL then give the list name

```
% dl
> raboof
Reading 'raboof'...
raboof>
```

In either case the new prompt will contain the list name. A "?" can be typed to obtain limited help.

To add a member, simply enter the ORName, using the syntax described in Section 3.2.3 at the "listname" prompt.

The "`p`" option is used to print the members of the list. If a DN is present, *only* the DN is printed. Each entry is preceded by a number. This number can be used to reference the entry when using the "`r`" — remove entry option, for example:

```
raboof> p

Owner: Colin Robbins

Submit Permissions,   Entities matching the OR pattern:
   X.400 Address: /C=GB/

Entry has no dl-policy

Members:
 1: Steve Hardcastle-Kille, Computer Science,
   University College London
 2: Colin Robbins, X-Tel Services Ltd
 3: Julian Onions, X-Tel Services Ltd

jnt-quipu> r
give username to be removed: 2
Remove Colin Robbins [y/n/d] ? n
no
```

The "`d`" option in the "`[y/n/d]`" prompt is used to display the directory entry ("`y`" for yes and "`n`" for no).

Alternatively an ORName could be given, but as UFN is invoked, it may take too long to find to correct user. Using numbers is quicker.

Attributes such as the list policy, and submission policy can be changed using the "`m`" option.

3.2.7 Maintaining a List

From time to time mail addresses will change. One of the advantages of the directory list is that such changes can to some extent be tracked [3] as long as the directory is kept upto date.

[3] If the DN changes you are still in trouble!

When accessing a particular list, the "`c`" and "`v`" commands to the DL program can be used to check or validate the list. The entry for each member of the list is checked for consistency. Each ORAddress is check to see if the local PP system thinks they are valid. If the ORName has a DN component, the given ORAddress is checked against the address given by the directory. Using the "`c`" option errors are reported, using the "`v`" an attempt is made to update bad ORAddresses.

You are advised to run "check" or "validate" on your lists regularly. Checking the lists can be done from *cron* using the "`-v`" option, for example:

```
dl -v raboof
```

This will warn (verbosely) of errors. The "`-q`" option is useful to quieten DL down.

You can check all lists this way, by not specifying a list name, for example:

```
dl -v -q -m
```

will (quietly) check all lists. The "`-m`" flag is used here to tell DL it is in management mode. Normally, the invoker of DL will be told of the errors in the list. When run at a site with a lot of lists, each with different managers, it is often more useful to mail the list managers of the errors, rather than mail the invoker of DL.

3.2.8 Printing a List

When DL prints the members of a large list, a pager such as *more* will be used to prevent the list scrolling past. Sometimes this can get in the way, for example if you want to pipe the output onto a printer. The "`-p`" flag to DL can be used to prevent this e.g.,

```
dl -p raboof
```

will cause DL to print the raboof list the the standard output, then exit.

Chapter 4

Miscellaneous Utilities

There are a number of small utility programs provided with PP that may be of interest to end users. These are described here.

4.1 A /bin/mail Replacement

PP comes with a very simple user interface for submitting messages. It has approximately the functionality normally assumed by */bin/mail* except that reading mail is not supported, only submission. That is, it takes a list of addresses on the command line and reads the body of the message on the standard input. By default all the addresses are marked as `To:` recipients. The command has a few options; each option stays in force until overriden.

- `-t`: The following arguments, up to the next flag, are `To:` addresses.
- `-c`: The following arguments are `Cc:` addresses.
- `-s`: The following arguments are text for the `Subject:` line.

This program is most useful in scripts and programs as a simple way of sending a message.

4.2 Message Status Checkup

PP provides a tool, *checkmail*, that connects to an MTA and finds out about messages related to the user.

There are two flags to alter *checkmail*'s behaviour.

-a: This flag prompts *checkmail* to find out about messages to and from the user. By default, *checkmail* only finds out about messages from the user.

-h MTAname: This flag prompts *checkmail* to connect to the named MTA. By default, *checkmail* attempts to connect to an MTA running on the local machine.

Here are two example displays from the *checkmail* tool: