PMDF User’s Guide
UNIX Edition

Order Number: U-5305-66-NN-U

September 2015

This document describes the UNIX user interfaces to Version 6.7 of the PMDF e-mail Interconnect family of products.

Revision/Update Information: This manual supersedes the V6.6 *PMDF User’s Guide, UNIX Edition*

Software Version: PMDF V6.7

Operating System and Version: Solaris SPARC or x86 V2.6, V8 or later; (SunOS V5.6, V5.8 or later);

Red Hat Enterprise Linux 4 update 8 or later on x86; (or other compatible Linux distribution)

Process Software, LLC
959 Concord Street
Framingham, MA 01701-4682 USA
Voice: +1 508 879 6994; FAX: +1 508 879 0042
info@process.com

Process Software, LLC (‘‘Process’’) makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Process Software reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Process Software to notify any person of such revision or changes.

Use of PMDF, PMDF-DIRSYNC, PMDF-FAX, PMDF-LAN, PMDF-MR, PMDF-MSGSTORE, PMDF-MTA, PMDF-TLS, PMDF-X400, PMDF-X500, PMDF-XGP, and/or PMDF-XGS software and associated documentation is authorized only by a Software License Agreement. Such license agreements specify the number of systems on which the software is authorized for use, and, among other things, specifically prohibit use or duplication of software or documentation, in whole or in part, except as authorized by the Software License Agreement.

Restricted Rights Legend

Use, duplication, or disclosure by the government is subject to
restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause at DFARS 252.227-7013
or as set forth in the Commercial Computer Software —
Restricted Rights clause at FAR 52.227-19.

Contents

EXAMPLES

FIGURES

TABLES

Preface

Purpose of This Manual

This manual describes the various user-level utilities and interfaces provided with the PMDF family of products. The intended audience is both users and system managers who will be using PMDF or any of the layered PMDF products.

Overview of This Manual

This manual consists of six chapters:

Chapter 1, Sending Mail Messages, explains how to send mail messages to PMDF using a variety of the more common user interfaces available for UNIX. This is the chapter to read if you want to learn how to send network mail.

Chapter 2, Receiving Mail Messages, explains briefly how to read mail messages delivered to you by PMDF.

Chapter 3, Message Filtering, Message Forwarding, and Vacation Notices, describes the web-based mail filtering facility.

Chapter 4, Password Change Utility, describes the web-based password change utility.

Chapter 5, Mail Server Commands, explains how to use a PMDF mail server to subscribe to mailing lists or obtain files made available through the server.

Chapter 6, Utilities, describes some PMDF shell command and interactive utilities.

Chapter 7, Notes for POP and IMAP Clients, describes some notes for users of remote POP or IMAP clients.

Availability

PMDF software products are marketed directly to end users in North America, and either directly or through distributors in other parts of the world depending upon the location of the end user. Contact Process Software for ordering information, to include referral to an authorized distributor where applicable:

Process Software, LLC
959 Concord Street
Framingham, MA 01701 USA
+1 508 879 6994
+1 508 879 0042 (FAX)
sales@process.com

Credits

Significant portions of PMDF Pine and some portions of the POP and IMAP servers are copyrighted as follows:

The following copyright and notice apply to the University of Michigan LDAP implementation.

The following additional notices apply to PMDF-TLS.

- This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
- This product includes some RSA Data Security, Inc., software.

1 Sending Mail Messages

This chapter focuses primarily on sending mail messages with `mail` and other user agents that interface to `sendmail`: `mail`, `mailx`, `mailtool`, `pine`, *etc*.

1.1 Sending Mail with mail, mailx, Pine, *etc*.

You should continue to use your regular `mail`, `mailx`, `mailtool`, `pine`, *etc*., mail user agent just as you always have in the past. These tools will invoke PMDF automatically as appropriate when sending mail.

1.1.1 Forwarding Mail and Delivery Mechanisms

You can have all of your mail automatically forwarded to a different mail address (or various other processing performed) by means of a `.forward` file; see Section 2.1 for an additional discussion.

Some similar functionality to forwarding is also available via the PMDF profile database; see Section 2.2.

1.1.2 Sending and Receiving Binary Files

A MIME-aware user agent such as pine can automatically send and receive binary attachments. However, if your mail user agent does not provide particularly useful facilities for sending binary data, then you may want to use the `pmdf send` shell utility to send such data. This utility can send multipart messages, and automatically encode, if necessary, binary attachment parts.

The `pmdf encode` and `pmdf decode` utilities can also be useful when you need to handle binary data with a mail user agent that does not have built in facilities for doing so.

1.2 Sending X.400 Mail (Solaris only)

If your system has PMDF-X400 installed, then you may be able to use it to send mail to X.400 subscribers on other systems. However, be warned that the information provided here is very general. In practice, the required information which you must supply to properly identify an X.400 addressee varies considerably from site to site as well as between X.400 service providers. You may find that you need very specific information in order to successfully address a message to a given X.400 subscriber (*e.g.*, what PRMD and ADMD to specify, whether or not initials are required, whether or not an organizational unit must be specified, *etc.*).

For sending mail to local X.400 users, you should consult your system manager or postmaster to determine the precise addressing conventions which you should use.

PMDF X.400 addresses have the format

```
"/attribute1=value1/attribute2=value2/.../"@X400-gateway-domain
```

where *X400-gateway-domain* is a domain name which was selected by your system manager when PMDF-X400 was installed. *attribute1*, *attribute2*, ... are any of the attribute names listed in Table 1–1 and *value1*, *value2*, ... are allowed values for the corresponding attributes. Consult your system manager for the proper domain name to use for *X400-gateway-domain*.

Table 1–1 X.400 Attribute Names Recognized by PMDF-X400

PMDF-X400 name	X.400 attribute
C	Country
ADMD	Administrative Domain
PRMD	Private Domain
O	Organization
OU	Organizational Unit (up to four of these fields are allowed)
S	Surname
G	Givenname
I	Initials
DD.xx	Domain Defined Attribute "xx"

Now, assuming that the *X400-gateway-domain* domain name is x400.example.com, then a sample X.400 address might be

```
"/g=John/s=Doe/o=Example Corporation/"@x400.example.com
```

This address specifies the recipient John Doe of the Example Corporation.

If any spaces or other special characters from the list of RFC 822 “specials” characters

```
        ( ) < > @ , ; : \ " . [ ]
```

appear amongst the attributes or values in the X.400 ORname portion of the address (the local-part of the address), then the entire local-part of the address must be enclosed in quotes, ". For instance, in the example above the space in `Example Corporation` has caused the entire local-part to require quoting.

An additional quoting is required when any of the characters =, $, or / appear in an attribute value. These characters must be quoted by prefixing each occurrence with a dollar sign, `$`. For instance, if "john doe a/s" is to be specified with an Organization attribute, then it must be specified as

```
/o=john doe a$/s
```

so that the "/" in "a/s" is not confused with the normal use of "/" as an attribute separator.

Note also that X.400 is limited in the characters that are allowed in attribute values to the characters defined as members of the "printable string" set. Characters that are used in RFC 822 (Internet) addresses but not allowed by X.400 are specified by multi-character substitutions in accordance with RFC 2156. These are particularly useful when X.400 originators need to specify a complete RFC 822 address in a `DD.RFC-822` domain-defined attribute. The substitutions are listed in Table 1–2.

Table 1–2 RFC 2156 Printable String Character Substitutions ORname in Fields

Desired character	X.400 multi-character
@	(a)
%	(p)
!	(b)
"	(q)
_	(u)
(	(l)
)	(r)

For example, to specify the RFC 822 address `doe@example.com` using the DD.RFC-822 attribute, the "@" must be specified as "(a)":

```
DD.RFC-822 = doe(a)example.com
```

Note: Since the (and) characters delimit comments in RFC 822 addresses, the use of printable string substitutions imposes the additional requirement that any address containing them must be enclosed in double-quotes. For example:

```
"/dd.rfc-822=doe(a)example.com/prmd=Gwy/admd=In/c=us/"@x400.example.com
```

Here, the double quotes are required to prevent the `(a)` from being discarded by the RFC 822 address processor through which this X.400 address is sent.

1.3 Delivery and Read Receipts

PMDF provides support for delivery receipts (a confirmation message is sent to you when your message reaches the recipient's mailbox).

Read receipts (a confirmation message is sent to you when your message is actually read by its recipient) are a matter for mail user agents (clients) that actually read the messages out of message stores, rather than mail transfer agents such as PMDF that deliver messages to message stores. PMDF passes read receipts through so that mail user agents that support read receipts may act upon them, should they want to.[1]

There are two separate issues concerning delivery and read receipts. First, a mechanism must exist with which to request delivery or read receipts. Your mail user agent may or may not allow you to request delivery or read receipts. Secondly, support must exist at the receiving end to recognize a request for a receipt — a receipt request — and generate the requested receipt. Again, PMDF provides this functionality. However, other mailers may not and thus it is not unusual for receipt requests to be ignored.

In general, delivery receipts should be requested according to the NOTARY mechanism specified in RFC 1891. In particular, when delivering to system mailboxes on the PMDF system or to PMDF MessageStore or PMDF popstore accounts, only NOTARY delivery receipt requests will be respected. For instance, to request a delivery receipt when using pine 4.44, first set the `enable-delivery-status-notification` option in the Setup Config menu; then when sending a message for which you want a delivery receipt, after pressing Ctrl/X (to Send), turn on the DSNOpts by pressing `D` (and then optionally turn on or off various of the notification options) before answering `Y` (to actually send the message).

Some PMDF delivery routes, such as delivery to LAN based e-mail systems such as cc:Mail, Microsoft Mail, GroupWise, or MHS based mail systems, also support delivery receipts requested via non-standard headers. If your user agent does not provide you a specialized way to request delivery receipts but does allow you to add your own headers, you may request a delivery receipt by adding a `Delivery-receipt-to:` header.

Read receipts should be requested using the standard `Disposition-notification-to:` header defined in RFC 2298. If your user agent does not provide you a specialized way to request read receipts but does allow you to add your own headers, you may request a read receipt by adding such a header. For instance, pine allows you to add any headers you want via the `customized-hdrs` Setup Config option. You may hence add a `Disposition-notification-to:` header to request a read receipt.

When delivering mail with a read receipt request, in some cases PMDF will convert the request into a delivery receipt request when it knows that the mail user agent to which the mail is being delivered cannot honor a read receipt request.

Warning: Never, ever request a receipt of any sort when posting a message to a mailing list. While some mailing lists properly block receipt requests, many do not. Should you accidentally request, for instance, a delivery receipt from a large mailing list, you will end up with hundreds, if not thousands, of receipts.

[1] Most mail user agents which support read receipts allow the recipient to block them. This is typically the default; *i.e.*, most mail reading programs will not generate read receipts unless the reader of the message expressly approves them.

1.4 sendmail Compatibility

On a PMDF system, a PMDF image is activated in place of `sendmail`; if you use a mail user agent that sends through `sendmail`, or if you invoke `sendmail` from the command line yourself, you will actually be running a PMDF image.

PMDF's `sendmail` replacement is designed for compatibility with mail user agent applications. However, it is not intended to be 100% compatible with `sendmail` when it comes to command line options for use by human users in submitting messages or management activities. PMDF's design has some fundamental differences from `sendmail`'s design; PMDF as a whole provides a safer implementation of all `sendmail`'s functionality, plus a great many additional capabilities. Not all `sendmail` command line options make sense for PMDF's `sendmail` replacement; the underlying functionality may be provided by PMDF in a way other than `sendmail` options.

Instead, for submitting messages from the command shell, see the `pmdf send` utility described in Chapter 6. For checking on any messages you have sent that are still in the PMDF queues, see the `pmdf qm` utility described in Section 6.3. Or for management activities such as testing address rewriting and general queue management, see the *PMDF System Manager's Guide*.

The command line options recognized by PMDF's `sendmail` replacement are shown in Table 1–3.

Table 1–3 PMDF's sendmail Replacement Command Options

Option	Usage
`-help`	Help; gives list of options supported
`-h`	Synonym for `-help`
`-t`	Use headers for envelope addresses
`-odq`	Enqueue only, with priority set to nonurgent
`-oee`	No error status returned to shell
`-f`*`user`*	Set From: address to *user*. In general, must be superuser, or in the `pmdf_world` group, to set *user* to other than one's own username, though users in the `pmdf_world_`*`username`* group are also allowed to specify `-f=`*`username`*.
`-bs`	SMTP dialog to `stdin`/`stdout` — all other options ignored
`-oi`	Line containing single dot (.) does not terminate message
`-i`	Synonym for `-oi`
`-r`*`user`*	Synonym for `-f`*`user`*
`--`	Any command line arguments after this option are ignored.
The following are always set when sending a message	
`-oem`	Mail error message back to user
`-om`	The sender address will not be stricken, if present, from the recipient list during alias expansion
`-m`	Synonym for `-om`
`-odb`	Background delivery (asynchronous)

Table 1–3 (Cont.) PMDF's sendmail Replacement Command Options

Option	Usage
The following are ignored without error messages	
`-v`	Set verbose mode
`-oo`	Old header format (with spaces instead of commas between addresses)
`-odi`	Interactive delivery
The following are not applicable, and generate an error	
`-bd`	Start SMTP server
`-bm`	Deliver mail
`-bp`	Print mail queue information
`-bt`	Test configuration

In addition, any other options not specified here are not supported and are ignored with an error message.

2 Receiving Mail Messages

By default, PMDF will deliver your new mail messages to your BSD mailbox. There are three common exceptions to this: (1) you have set up a `.forward` file directing your messages elsewhere, (2) you have set a PMDF profile database entry directing your messages elsewhere, or (3) your system manager has set a PMDF profile database entry directing your messages elsewhere.

2.1 The .forward File

PMDF normally looks for your `.forward` file in your home directory. However, your PMDF system manager may have chosen to configure PMDF to look in a different location for user `.forward` files; check with your PMDF system manager if you have any question about where your `.forward` file should reside.

The format for a `.forward` file is one or more lines, each line containing one or more comma-separated recipient entries. A recipient entry may take the following forms.

- *user@domain* directs a copy of the message to the specified address.
- \your-account-name delivers a copy to you "normally" (in addition to any other forwarding specified by other entries).
- /*directory*/*path*/*filename* appends a copy of the message to the specified file. (The specified file must be writeable by you.)
- +/*directory*/*path*/*filename* appends a copy of the message to the specified file using digest-like boundary markers between the messages. (The specified file must be writeable by you.)
- |*command* where *command* is a shell command will execute that command as you, with your access and privileges, with standard input coming from the message being delivered.
- |*command args* where *command args* is a shell command with arguments will execute that command as you, with your access and privileges, with standard input coming from the message being delivered.

Lines beginning with an exclamation point character, `!`, are considered to be comment lines.

Note that when executing certain actions of your `.forward` file, PMDF becomes you. In particular, any scripts executed by your `.forward` file will be executed as you, with your access and privileges.

Example 2–1 shows a sample `.forward` file for a user jdoe who wants one copy of his messages delivered normally, one copy sent to another of his accounts, one copy sent to his pager, one copy filed in the file `thismonthsmailarchive`, and also wants to pipe the message through the `procmail` utility.

Example 2–1 Sample `.forward` file for user `jdoe`

```
\jdoe, John.Doe@system2.example.com, John.Doe@pager.example.com
/usr/users/jdoe/thismonthsmailarchive
"|/usr/bin/procmail jdoe"
```

2.2 The PMDF User Profile Database

If your system manager has configured PMDF user profile database options, then you can select among those options for delivering your mail. Note that your system manager may have selected a delivery option for you or a default delivery option for all users on your system, so you may even have a user profile option set without realizing it.

To see what options, if any, your system manager has configured for your site, issue the command

```
% pmdf profile show method -all
```

If your system manager has configured any options, they will be listed in the output. For instance, if your system manager has configured two options, regular BSD mailbox delivery and delivery to a MailWorks mailbox, you might see output such as:

```
% pmdf profile show method -all
Method BSD is defined as: /var/spool/mail/%s
Method DMW is defined as: |/usr/bin/inetgrecv %s
```

To see what your own delivery method is set to, if it is set at all, issue the command

```
% pmdf profile show delivery
```

To set your delivery method, issue the command

```
% pmdf profile set delivery method-name
```

where *`method-name`* is the name of a method defined by the system manager, listed in the output of a `pmdf profile show method -all` command. For instance,

```
% pmdf profile
profile> show method -all
Method BSD is defined as: /var/spool/mail/%s
Method DMW is defined as: |/usr/bin/inetgrecv %s
profile> set delivery DMW
profile> exit
```

See Section 6.2 for additional discussion of using the `pmdf profile` utility.

2.3 Receiving Mail in mail, mailx, *etc*.

New messages delivered by PMDF to your mailbox may be read with a variety of programs including `mail`, `mailx`, Pine, *etc*. You read these messages as you would any other mail message; no special action is required. Messages delivered by PMDF will have valid `From:` addresses and can be replied to with the standard reply commands.

2.3.1 Receiving Binary Files

If receiving a binary file using a MIME aware user agent such as Pine, the binary file should be extractable and usable.

If you are using a non-MIME aware user agent and receive a MIME encoded binary file, note that the `pmdf decode` utility may be used to decode MIME encodings.

2.3.2 Header Lines in Received Messages

There are a large number of fields which can appear in the header of a message. These include familiar things like `From:`, `To:`, `Subject:`, and `Return-Path:` and obscure things like `Encrypted:`, `Resent-Message-ID:`, and `Resent-reply-to:`. In particular, if the user agent you use to read your messages does not understand MIME message formatting, you are likely to see headers such as `MIME-version:`, `Content-type:`, and `Content-transfer-encoding`; MIME-aware user agents use these headers internally rather than presenting them to you, the reader.

When delivering to something which does not understand MIME, there are two choices: either discard the header lines that the user agent does not understand, or insert them somewhere in the body of the message. Neither choice is completely satisfactory — deleting header lines causes the loss of valuable information but inserting header lines into the message text may interfere with extract commands. PMDF, unless configured otherwise by the system manager, uses the latter approach: PMDF preserves additional header lines by merging them into the text of the message.

2.4 Receiving Mail with Pine

The pine mail user agent developed at the University of Washington can read messages from the BSD mailbox, and it can also be used as an IMAP client or POP3 client to read messages from mailboxes on remote systems.

If only local mail is to be accessed (your BSD mailbox on the same system), no network connection is created, and no IMAP server or POP server is needed on the local system. If you have been using pine in this way you should continue to use it as always.

If you want to connect via pine to read mail files on a remote system, then there must be TCP/IP connectivity between your system where you are running pine and the remote system. The remote system must be running an IMAP daemon or server such as the PMDF IMAP server, or a POP daemon or server such as the PMDF POP server. [1]

Pine is also an NNTP (Network News Transport Protocol, RFC 977) client, and can be used to read news from NNTP servers like the ANU News program or many common UNIX NNTP servers. Of course, you have to know the name of one such system before you can use this capability. Ask your system or network manager for help.

The pine view is that messages are stored in folders, and folders are stored in folder collections. Folder collections may be physically located on the local system, or on any remote system with an IMAP server. Regardless of what system a folder collection is physically located on, a pine user sees it as just another folder collection: a pine user can read messages in any of their folder collections and can save (move) messages between different folder collections. See Section 2.4.1 below for more details on folders and folder collections.

Or a read-and-delete-only pine folder can correspond to the "new messages" folder on a remote system with a POP3 server. (The POP3 protocol does not provide access to multiple folders—it only provides access to the "new" messages, usually those in a special "new" sort of folder. The POP3 protocol also does not allow for moving messages into a folder.)

2.4.1 Pine Folder Collections

A *folder collection* is a folder specification for a collection of folders on one system. For example, it can be all of your UNIX mail folders (mail files) on a system called foo.bar.com, or all of your UNIX mail folders (mail files) whose names begin INFO-, or it can be all of your VMS MAIL folders on a remote OpenVMS system. You can access multiple different folder collections from within pine.

By default, pine knows only about the local folder collection, corresponding to your BSD mailbox. The use of additional folder collections is controlled by the `folder-collections` option in your pine resource file. Normally, this option is set from within pine by using the `SETUP` menu and then selecting the `L collectionList` menu. However, the option can also be set by manually editing your pine resource file.

[1] To find out more about IMAP, read the IMAP4rev1 specification stored in the file `/pmdf/doc/rfc2060.txt`; to find out more about POP, read the POP3 specification stored in the file `/pmdf/doc/rfc1939.txt`.

2.4.1.1 folder-collection Option Syntax

The setting of the `folder-collections` option may be a list of values, where each value specifies a folder or folders on the local system or accessible via an IMAP server, or specifies the new mail folder accessible via a POP3 server. Folders on the local system or accessible via an IMAP server are specified using the format:

optional-label {*imaphost*}*optional-file*[*view*]

or

optional-label {*imaphost*:*port*/user=*username*}*optional-file*[*view*]

optional-label is a label which will be displayed by pine in place of the full name of the folder collection.

The optional field *imaphost* is the name of a host where the mail file resides. *imaphost* may be any system which has an IMAP server.

The optional *port* specification may be included if you want to connect to a port other than the default (for IMAP) of 143.

The optional *username* may be included if you want to log in to the *imaphost* under a different account name.

The optional field *optional-file* is the file specification of a mail file. If *optional-file* is omitted but *imap-host* is specified, then the default mail file on the remote *imaphost* system will be used. If neither *optional-file* nor *imap-host* is specified, then your local default mail file will be used.

When connecting to an IMAP server on a UNIX system, the *optional-file* part of the specification is usually not used; instead, one simply specifies the mail folder(s) (mail file(s)) to access by specifying them as the *view* part of the value.

When connecting to a OpenVMS based PMDF IMAP server, however, the *optional-file* part of the specification must have the format

#*disk*:<*directory*>*mailfile.mai*#

where *disk*, *directory*, and *mailfile.mai* specify the full path, disk, directory, and file name, to the mail file. For instance, to select the mail file `MEMOS.MAI` of `DISK$USER1:[BOB]`, you would specify

```
#DISK\$USER1:<BOB>MEMOS.MAI#
```

Finally, the *view* field controls which folders from the mail file are part of the collection. If specified as being empty, `[]`, then all folders from the mail file are treated as part of the collection. Wild cards may be used to select folders matching a pattern. For example, `[INFO-*]` would select all folders beginning with the string `INFO-` from the mail file. Again, note that folder names are considered to be case sensitive.

For POP3 access to a new mail folder on a remote system, the format is:

```
"foldername" {pop3host/POP3}INBOX
```

or

```
"foldername" {pop3host/POP3/USER=username}INBOX
```

where *foldername* is the name by which pine will refer to the folder, *pop3host* is the name of the system running the POP3 server, and *username* is the name under which to log in to the remote POP3 server.

2.4.1.2 Example folder-collections Option Setting

An example of setting the `folder-collections` option in your pine resource file, normally `.pinerc`, to a list of several folder collections is:

```
folder-collections=local [],
 archive [/archive/mail],
 remoteVMS {vax.example.com}#DUA2:<JONES.MAIL>MAIL.MAI#[INFO*]
 remoteUNIX {sun.example.com}mail/[]
```

In the above example, four collections with the names local, archive, remoteVMS, and remoteUNIX are created. local consists of all folders in the local default mail file; archive consists of all folders in the mail file `~/archive/mail`; remoteVMS consists of all folders whose name begin with `INFO` in the mail file `DUA2:[JONES.MAIL]MAIL.MAI` on the remote host vax.example.com; and remoteUNIX consists of all folders from the mail directory `mail/` on the remote system sun.example.com.

2.5 Subaddresses

You may include extra information in your mail address. This extra information is referred to as "subaddress". Any material following a plus, +, in your address is considered to be a subaddress. For instance, "test" in the address bob+test@example.com is a subaddress.

For BSD mailbox users, subaddresses do not in any way hinder your mail from being delivered to you. If your address is bob@example.com, then messages sent to bob+test@example.com will be delivered to you just as messages to bob@example.com would be.

For PMDF MessageStore users, subaddresses control which folder your messages are delivered to. If your address is bob@example.com, then messages sent to bob+test@example.com will be delivered to the folder named "test" of your bob MessageStore account.

Subaddresses are useful even for BSD mailbox users for whom they are merely "cosmetic" information. One common use of subaddresses is to track where mail is coming from. Let us again assume that your mail address is bob@example.com. Now, if you subscribe to the info-unix mailing list using the address bob+info-unix@example.com, then mail to you from that list should show that subaddress somewhere in the message header. Indeed, whenever you get mail addressed to the subaddress info-unix, then you know that the mail came from that list.

Finally, subaddresses can serve as simple visual cues to inform you about where a message originated. You may elect to always include a subaddress in your address when you tell it to other people. It is then possible to figure out where a message came from just by what subaddress your `To:` address shows.

3 Message Filtering, Message Forwarding, and Vacation Notices

A common goal is to filter incoming messages, perhaps sending an automatic response, forwarding the messages to another account, or automatically rejecting or discarding some based upon material in the messages' headers or bodies. One way to do this is via a program such as `procmail`; check with your system administrators for what facilities may be available on your system.

If your messages are delivered to the native message store (Berkeley mailbox) or to a PMDF popstore or PMDF MessageStore account, your system administrators may have chosen to enable another option: PMDF message filtering. If your system administrators have chosen to enable it, PMDF provides a web-based interface through which you can specify a vacation notice, specify a forwarding address, and construct and manage your own message screening rules. These are collectively known as ***mailbox filters***.

3.1 Mailbox Filter File

By default, you have no mailbox filters unless your system administrator has chosen to set them up for you. If mailbox filtering has been enabled for your account, when you use the web-based interface, a mailbox filter is created for you. Your mailbox filters are stored in a mailbox filter file. The location of that file is site-configurable. In most cases, that location is in a directory that is not directly accessible by non-privileged users, and all modifications to your mailbox filters are done using the web interface.

If your system administrator has set up your mailbox filter file to be accessible by you, you may create or modify the filter file using any text editor. The mailbox filter file is a text file containing commands in the SIEVE language with some extensions. See the *System Manager's Guide*, Chapter 16, for more information about SIEVE.

Warning: If you edit your mailbox filter file manually, you *cannot* use the web interface any longer. The web interface can only read filter files it has written itself. You can use the web interface to create an initial mailbox filter file, and then edit it manually, but not vice-versa.

3.1.1 Checking Your Changes

Note: After you have made changes to your mailbox filter file, it is important for you to verify that it is working correctly, especially if you have edited it manually. If your filter file is not working, for example if it has a syntax error, your mail delivery could be interrupted.

The easiest way to check your mailbox filter file is to send yourself mail. If your mail gets to your mailbox successfully, then there is nothing wrong with your filter file.

Your filter file can also be verified by your system administrator using the command:

```
# pmdf test -rewrite -filter <your-mailbox>
```

3.2 Web Interface

In order to use the web-based interface for setting up message filters, a vacation notice, or a forwarding address, you must have a web client and TCP/IP access to the PMDF system. Your messages must also be delivered to the native message store (Berkeley mailbox) on the PMDF system, or to a PMDF popstore or PMDF MessageStore account on the PMDF system.

The web form asks you for your e-mail address and your password; you need to provide this information in order to set up or change your mailbox filters.

To connect to the interface with your web browser, you normally open the URL:

```
http://host:7633/mailbox_filters
```

In place of *host*, use the actual IP host name of the system running PMDF, on which your messages are delivered. Your system administrator may have chosen to configure the web interface port to be a port other than 7633; if so, then you need to specify that other port number in place of 7633 in the above URL. Check with your system administrator if you are not sure of the exact URL to use.

Once connected to the introductory web page, links to help and various mailbox filtering activities may be followed.

3.2.1 Web Interface Features

The web interface allows you to set up eight distinct message filters: four to identify messages to always keep, the ***Accept filters***; four to identify messages to always throw away, the ***Discard filters***. The Accept and Discard filters operate on envelope and header source addresses, header destination addresses, and phrases or words appearing in the Subject: header line or body of the message. The eight filters are thus known by the names Accept From, Accept To, Accept Subject, Accept Body, Discard From, Discard To, Discard Subject, and Discard Body.

The web interface also allows you to set up a forwarding address. When you have a forwarding address set up, all of your mail that you have decided to keep with your Accept filters will be sent to that address instead of being delivered to your local account. Note that the Accept and Discard filters are applied first, and the vacation notice (if any) is also sent first, before the message is forwarded.

The web interface also allows you to set up a vacation notice. Set up a vacation notice when you want to send an automatic reply to mail messages that you receive. The reply notifies the sender that you are on vacation or otherwise away for an extended period of time and may not respond to your mail until you return. The web interface allows you to enable or disable the vacation notice feature, to specify the subject and text that is included in the vacation notice, and to set up some advanced options.

PMDF keeps a history of which addresses it has sent the vacation notice to, and does not send another vacation notice to that same address unless

- you change the text or subject of your vacation notice
- you enable the vacation notice feature after it has been disabled
- the number of days that you specify in the web interface has passed

Note that PMDF will not send the vacation notice if it determines that the message was received through a mailing list.

3.2.2 Example Web Page Displays

The figures below show samples of some of the mailbox filter pages provided by default with PMDF. Note that different web browsers may display pages a bit differently, and your PMDF manager may have customized the mailbox filter web pages for your site.

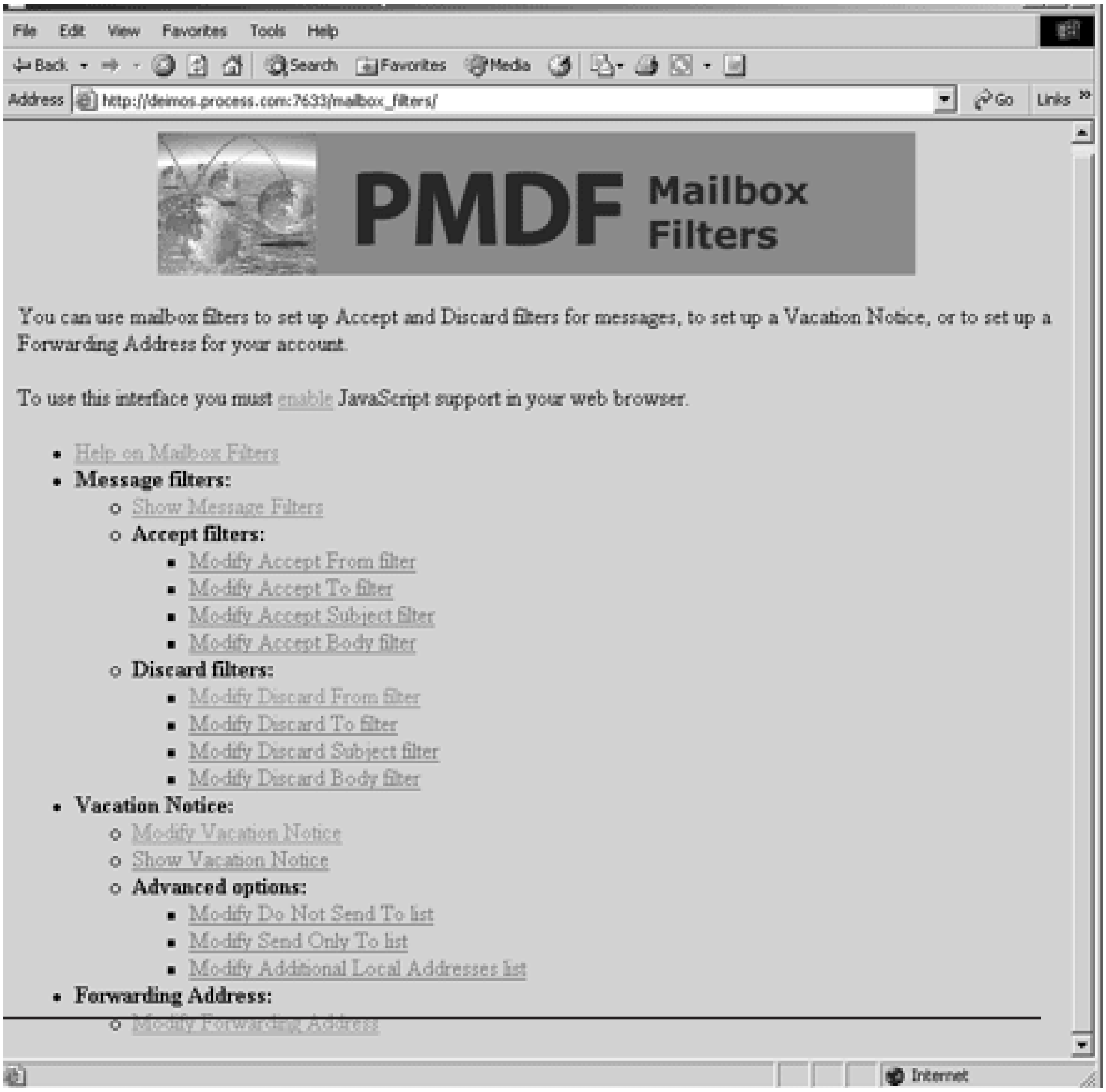
Figure 3–1 shows a sample Mailbox Filters home page—the first page you will see when you connect to the mailbox filter URL. This home page has links to other pages which implement each of the mailbox filters features: Accept filters, Discard filters, vacation notice, and forwarding address.

For instance, you may click on the "Modify Discard From filter" link displayed on the mailbox filter home page to move to the `Discard From` filter page, shown in Figure 3–2. (If this is is the first page you go to from the mailbox filter home page, note that you will be asked to authenticate yourself before your `Discard From` filter page will be shown; a dialogue box will pop up asking you to provide your e-mail address and password.) The example in Figure 3–2 is for a user who has configured his mailbox filter to reject all messages from hotmail.com or cyberpromo.com.

As another example, clicking on the "Modify Vacation Notice" link displayed on the mailbox filter home page moves you to the `Vacation Notice` page, shown in Figure 3–3.

Help is available on the various mailbox filtering pages; for instance, clicking on the `Help` button in the lower right corner of the `Discard From` filter page will move you to a help page describing how to use the `Discard From` filters; an excerpt of that help page is shown in Figure 3–4.

Figure 3–1 Mailbox Filter Home Page



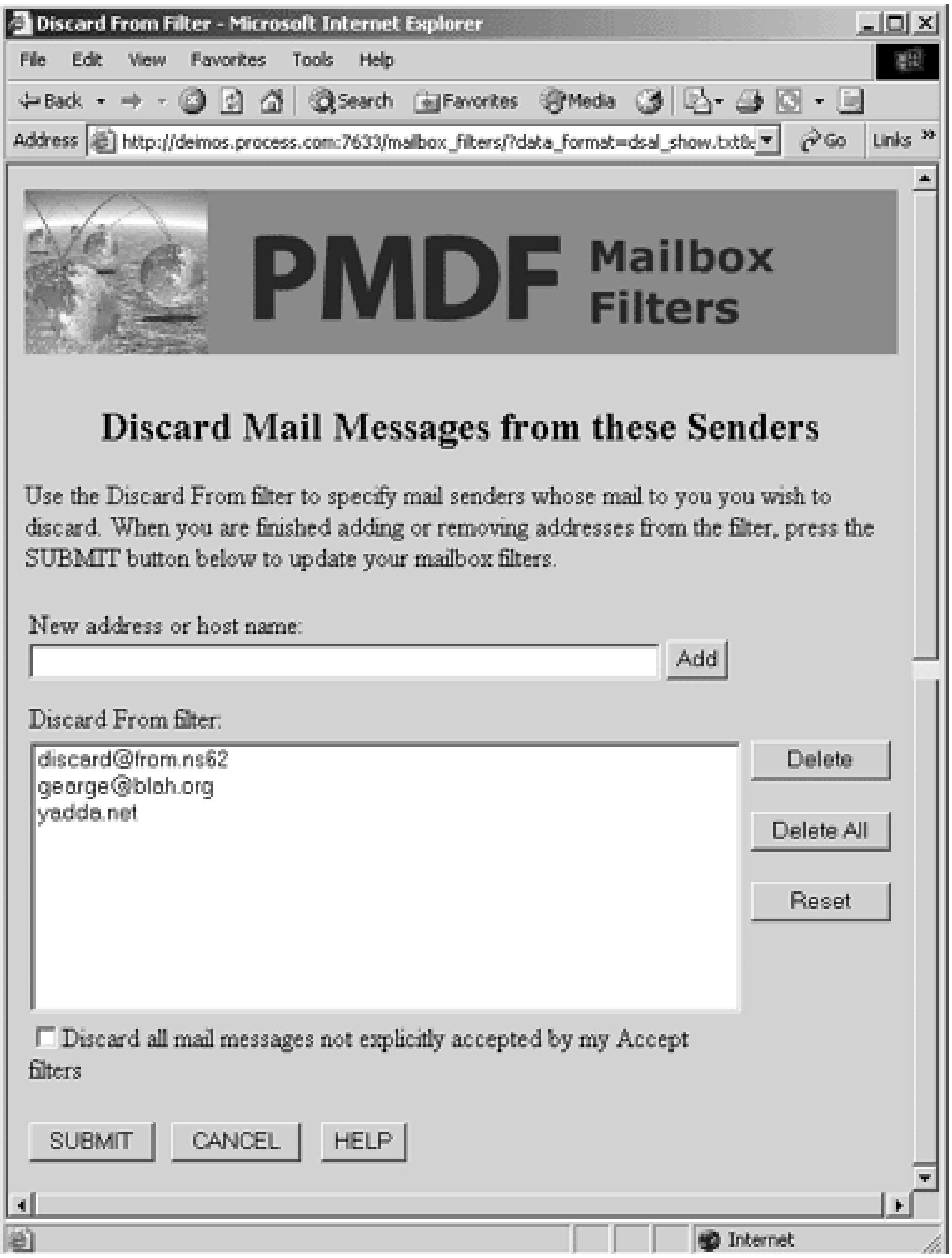
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://deimos.process.com:7633/mailbox_filters/
Go Links
PMDF Mailbox Filters
Internet

You can use mailbox filters to set up Accept and Discard filters for messages, to set up a Vacation Notice, or to set up a Forwarding Address for your account.

To use this interface you must enable JavaScript support in your web browser.

- Help on Mailbox Filters
- **Message filters:**
  - Show Message Filters
  - **Accept filters:**
    - Modify Accept From filter
    - Modify Accept To filter
    - Modify Accept Subject filter
    - Modify Accept Body filter
  - **Discard filters:**
    - Modify Discard From filter
    - Modify Discard To filter
    - Modify Discard Subject filter
    - Modify Discard Body filter
- **Vacation Notice:**
  - Modify Vacation Notice
  - Show Vacation Notice
  - **Advanced options:**
    - Modify Do Not Send To list
    - Modify Send Only To list
    - Modify Additional Local Addresses list
- **Forwarding Address:**
  - Modify Forwarding Address

Figure 3–2 Discard From Filter


Discard From Filter - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://deimos.process.com:7633/mailbox_filters/?data_format=dsal_show.txt&
Go Links
PMDF Mailbox Filters

Discard Mail Messages from these Senders

Use the Discard From filter to specify mail senders whose mail to you you wish to discard. When you are finished adding or removing addresses from the filter, press the SUBMIT button below to update your mailbox filters.

New address or host name:

Add

Discard From filter:

discard@from.ns62
george@blah.org
yadda.net

Delete

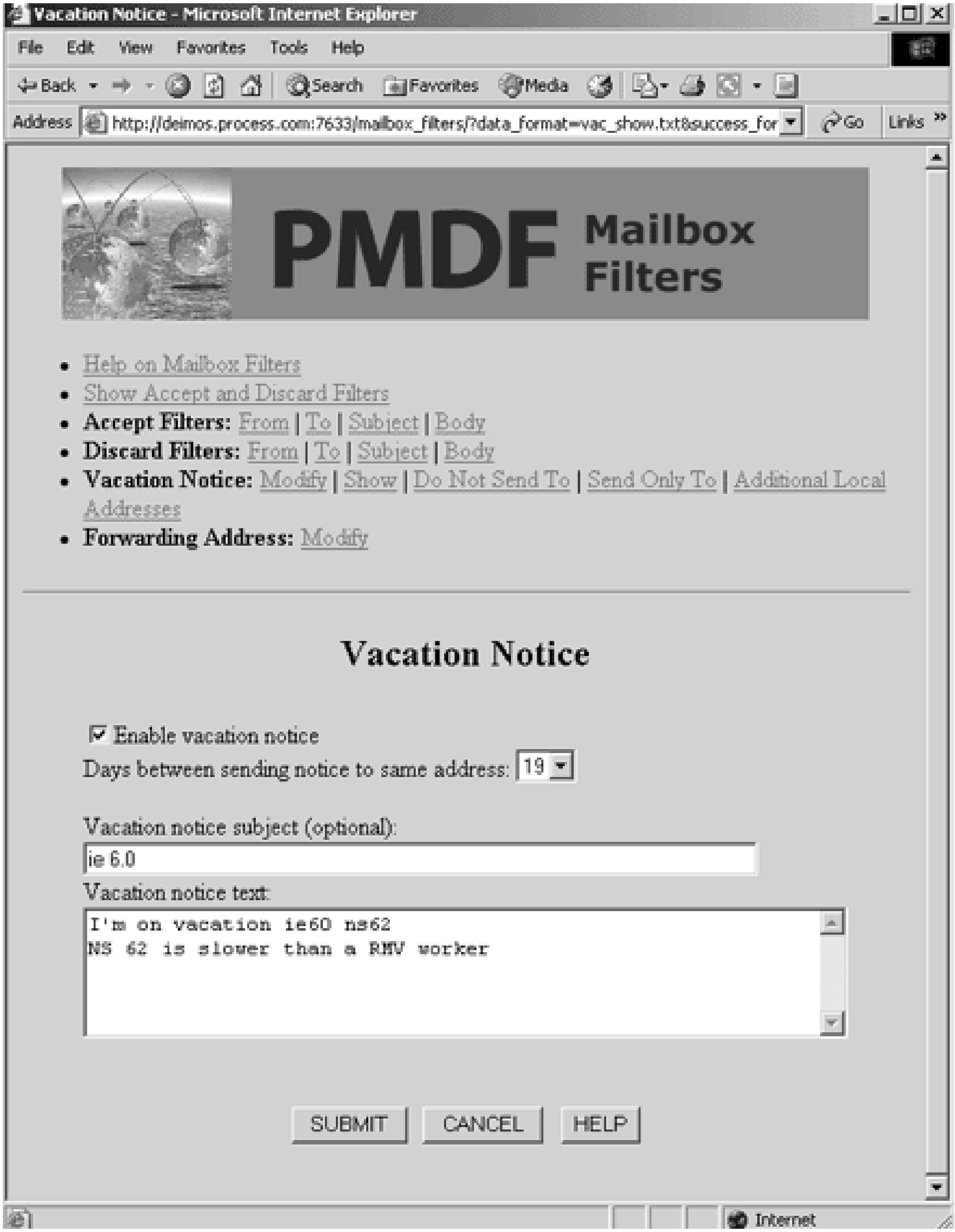
Delete All

Reset

□ Discard all mail messages not explicitly accepted by my Accept filters

SUBMIT CANCEL HELP

Figure 3–3 Vacation Notice


Vacation Notice - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://deimos.process.com:7633/mailbox_filters/?data_format=vac_show.txt&success_for
Go Links
PMDF Mailbox Filters
Internet

- Help on Mailbox Filters
- Show Accept and Discard Filters
- **Accept Filters:** From | To | Subject | Body
- **Discard Filters:** From | To | Subject | Body
- **Vacation Notice:** Modify | Show | Do Not Send To | Send Only To | Additional Local Addresses
- **Forwarding Address:** Modify

Vacation Notice

☑ Enable vacation notice

Days between sending notice to same address: 19

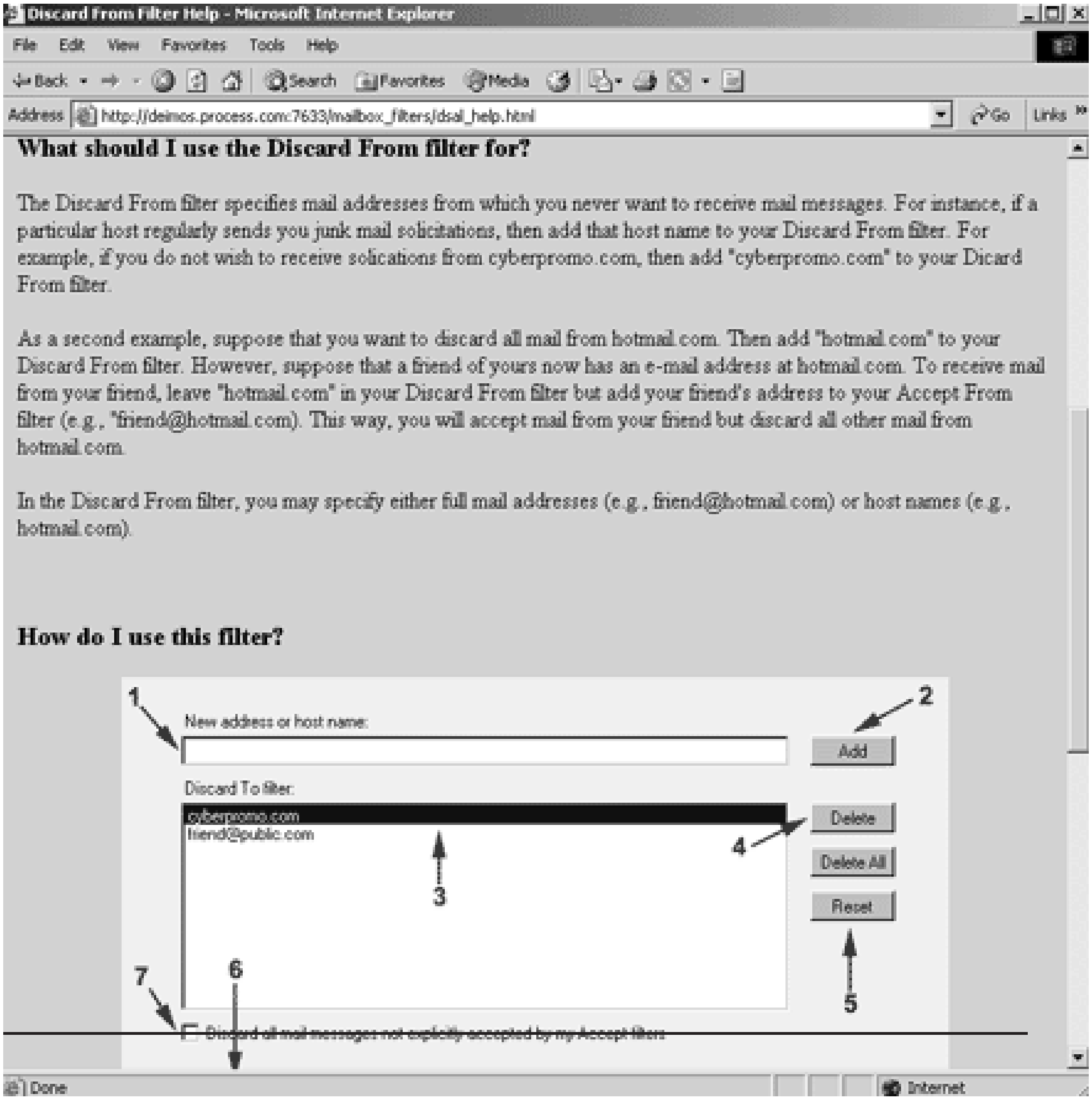
Vacation notice subject (optional):
ie 6.0

Vacation notice text:

```
I'm on vacation ie60 ns62
NS 62 is slower than a RMV worker
```

SUBMIT CANCEL HELP

Figure 3–4 Excerpt of the Discard From Filter Help Page



Discard From Filter Help - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://deimos.process.com:7633/mailbox_filters/dsal_help.html
Go Links
1
2
3
4
5
6
7
Add
Delete
Delete All
Reset
Done
Internet

What should I use the Discard From filter for?

The Discard From filter specifies mail addresses from which you never want to receive mail messages. For instance, if a particular host regularly sends you junk mail solicitations, then add that host name to your Discard From filter. For example, if you do not wish to receive solications from cyberpromo.com, then add "cyberpromo.com" to your Dicard From filter.

As a second example, suppose that you want to discard all mail from hotmail.com. Then add "hotmail.com" to your Discard From filter. However, suppose that a friend of yours now has an e-mail address at hotmail.com. To receive mail from your friend, leave "hotmail.com" in your Discard From filter but add your friend's address to your Accept From filter (e.g., "friend@hotmail.com). This way, you will accept mail from your friend but discard all other mail from hotmail.com.

In the Discard From filter, you may specify either full mail addresses (e.g., friend@hotmail.com) or host names (e.g., hotmail.com).

How do I use this filter?

New address or host name:

Discard To filter:

cyberpromo.com
friend@public.com

☐ Discard all mail messages not explicitly accepted by my Accept filters

4 Password Change Utility

If your system administrators have chosen to enable it, PMDF provides a web-based interface through which you can change the password on your account. This applies to all accounts, whether they are native message store (Berkeley mailbox) accounts, or popstore accounts, or MessageStore accounts, or if your username and password is located elsewhere for example in the PMDF password database.

To use the web-based password change utility, you must have a web client and TCP/IP access to the PMDF system. To connect to the interface with your web browser, you normally open the URL:

```
http://host:7633/chng_pwd
```

In place of *host*, use the actual IP host name of the system running PMDF. Your system administrator may have chosen to configure the web interface port to be a port other than 7633; if so, then you need to specify that other port number in place of 7633 in the above URL. Check with your system administrator if you are not sure of the exact URL to use.

Once connected to the web page, you must fill in your username and the string that you want to change your password to. Make sure that you type the new password in twice, correctly, for verification. When you click on the "Change Password" button, your web browser will prompt you to enter your username and existing password. Type these in to authenticate yourself. PMDF will then modify your password as requested.

5 Mail Server Commands

PMDF provides a combined mail and list server, referred to in this document as a “mail server”. Mail servers are used to distribute files via e-mail and allow users to subscribe or unsubscribe from mailing lists. If your system manager has configured a mail server at your site, then you can query the server via e-mail to determine what files and mailing lists are available.

5.1 Sending Commands to the Server

Commands directed to a mail server take the form of a mail message addressed to

```
mailserv@mail-server-host
```

where *mail-server-host* is the host name of the machine running the mail server. You need to obtain this name from your system manager. The text of the message contains mail server commands, one command per line.

Mail server command names are not case sensitive themselves. MAILSERV commands preserve the case of specified file names.

For example, suppose the address of a mail server is mailserv@example.com. To obtain a help message from the server as well as a list of the available files and mailing lists, you would send a message much like the one shown in Example 5–1.[1]

Example 5–1 Sending Commands to a Mail Server

```
% mail mailserv@example.com
HELP
INDEX
LISTS
CTRL/D
%
```

After a short while, you will receive back three messages from the mail server: the first message in response to the `HELP` command, the second in response to the `INDEX` command, and the third in response to the `LISTS` command. If your command specifications are in error, then the mail server will send you an error notification.

[1] Note that PMDF mail servers, and many others too, ignore the `Subject:` line in messages sent to the server. Do not attempt to place commands in the `Subject:` line; they will be ignored.

Typically, mail from the mail server will have a reply address which differs from the address you use to `send` commands to it. This is intentional and is done to prevent potential mail loops. One consequence of this is that you cannot direct further commands to the server by replying to messages from it. You must always initiate a new message with the `send` command; you cannot use the reply command.

A brief description of the available commands is given in Table 5–1; complete descriptions begin in Section 5.2.

Table 5–1 Summary of Mail and List Server Commands

Command	Description
DIRECTORY	Obtain directory listing of available files
DIRECTORY/LIST	Obtain directory listing of available mailing lists
ENCODING	Set default file transmission encoding
END	Terminate processing, accept no additional commands
EXIT	Same as END
FINISH	Same as END
HELP	Retrieve the server-specific help information
INDEX	Retrieve the index of available files
LISTS	Retrieve the index of available mailing lists
MAXIMUM	Set maximum message size; large messages will be divided into several messages, each smaller than this size
MODE	Set the default file reading mode
PURGE/LIST[1]	Purge comment lines (such as unsubscribed addresses) from the membership list
QUIT	Same as END
SEND	Retrieve the specified files
SEND/LIST[1]	Retrieve the membership list for a given mailing list
SEND/LIST/COMMENTS[1]	Retrieve the membership list for a given mailing list, including members' RFC 822 comment fields
SEND/LIST/NOCOMMENTS[1]	Retrieve the membership list for a given mailing list, stripping members' RFC 822 comment fields
STOP	Same as END
SUBSCRIBE	Subscribe to a mailing list
UNSUBSCRIBE	Unsubscribe from a mailing list

[1] Use of this command is generally restricted to authorized users.

5.2 Available Commands

The remainder of this chapter documents the available mail server commands. Please note that these commands are those accepted by the PMDF mail server. While there is a fair amount of commonality amongst the commands accepted by many of the different mail servers, differences also exist. Do not expect these commands to work with other mail servers and do not expect other mail server commands to work with PMDF mail servers. When you are unsure of what sort of mail server you are dealing with, the first order of business should be obtaining help information for that server. Often this can be done by sending the command `HELP` to the server.

CONFIRM

Confirm a command from a previous message.

SYNTAX

CONFIRM *cookie*

PARAMETERS

cookie
Required cookie string to confirm the command.

DESCRIPTION

The `CONFIRM` command is used to confirm for MAILSERV the execution of a command from a previous message.

That is, for security your system administrators may have configured MAILSERV to require confirmation of certain commands. If you receive a message from MAILSERV saying that you need to send a

```
CONFIRM cookie-string
```

message back to MAILSERV in order for it to perform some particular command you previously requested, then if you want that command executed you must send back exactly

```
CONFIRM cookie-string
```

where *`cookie-string`* is the exact string MAILSERV tells you to send for that command.

Note that you should send a new message back to MAILSERV containing the required `CONFIRM` command, rather than simply resending or bouncing MAILSERV's own message back to MAILSERV (to ensure that you hear about any errors in processing your `CONFIRM` command).

Note that if you receive a message from MAILSERV talking about confirming a command that you did *not* send yourself, then that may mean that someone is attempting to masquerade as you in e-mail and you may want to take this up with your system administrators.

ERROR MESSAGES

```
%MAILSERV-F-NOCOOKIE, There is no confirmation-pending command labeled
```

There was no command corresponding to such a cookie string awaiting confirmation. Check that you entered the cookie correctly.

DIRECTORY

Obtain a directory listing of the available files.

SYNTAX

DIRECTORY *[file-spec]*

PARAMETERS

file-spec
Optional file name specification indicating which files to obtain a directory listing of. A directory specification may be used. The usual shell wildcard characters, `*` and `?`, are supported.

DESCRIPTION

The `DIRECTORY` command provides a directory listing of the available files. The listing is returned to you as a mail message.

The **file-spec** parameter is optional and, if omitted, defaults to `*`. If you are unsure of what to use, omit the parameter and send simply the command

```
DIRECTORY
```

This will provide you with a list of the files and directories in the top-level directory of the mail server. You can then use this information to refine your queries; *e.g.*, investigate the contents of an intriguing directory,

```
DIRECTORY games/*
```

ERROR MESSAGES

`%MAILSERV-W-NOFILES, no files found`

The supplied file specification does not match any available files.

`%MAILSERV-F-NOFILESERV, file service is not enabled`

The mail server is not configured to operate as a file server.

`%MAILSERV-W-WRITEERR, file writing error`

An error occurred while the server was producing your directory listing. Try resending the command at a later time.

DIRECTORY/LIST

Obtain a listing of the available mailing lists.

SYNTAX

DIRECTORY/LIST *[list-spec]*

PARAMETERS

list-spec

Optional mailing list specification indicating which mailing lists to obtain a listing of. The usual shell wildcard characters, `*` and `?`, are supported.

DESCRIPTION

The `DIRECTORY` command provides a listing of the available mailing lists.

The **list-spec** parameter is optional and, if omitted, defaults to `*`. Generally, there is no need to use this parameter unless you are interested in a specific mailing list. For instance, if you merely want to know if there is a mailing list about zeugmes, you might use the command

```
DIRECTORY/LIST *ZEUGME*
```

This will provide you with the names of any mailing lists which contain the phrase “zeugme” in them. Note that just because a mailing list is available does not necessarily mean that you can subscribe to it. The site may have established restrictions governing who may or may not subscribe to some or all mailing lists.

ERROR MESSAGES

`%MAILSERV-W-NOLISTS, no lists found`

The supplied mailing list specification does not match any available mailing lists.

`%MAILSERV-F-NOMAILLIST, mailing lists are not enabled`

The mail server is not configured to operate as a list server.

`%MAILSERV-W-WRITEERR, file writing error`

An error occurred while the server was producing your listing of mailing lists. Try resending the command at a later time.

ENCODING

Specify the file encoding to use.

SYNTAX

ENCODING *encoding*

PARAMETERS

encoding

Required parameter specifying the file encoding to use. The available encodings are: `8BIT`, `7BIT`, `BASE32`, `BASE64`, `CBASE64` (gzip compressed `BASE64`), `BASE85`, `BINHEX` (encoding only, not the BINHEX file format), `BTOA`, `HEXADECIMAL`, `PATHWORKS`, `QUOTED_PRINTABLE`, `UUENCODE`, and `CUUENCODE` (gzip compressed `UUENCODE`).

DESCRIPTION

Binary files cannot be transmitted directly as electronic mail; they must first be encoded into a "printable" format. This, of course, means that they must be decoded upon receipt. The `ENCODING` command is used to specify the encoding to be applied to files requested with the `SEND` command. When selecting an encoding, be sure to select an encoding which you can decode. If your mail is handled by PMDF, then you can decode any of the encodings offered by PMDF mail servers.

The encoding specified with the `ENCODING` command applies to all subsequent SEND commands in the same message. It may be overridden with a subsequent `ENCODING` command or, on a per command, basis with the `SEND` command's `/ENCODING` qualifier. And, of course, encodings established in previous messages sent to the server have no effect on subsequent messages which you might send.

The `BASE64` and `QUOTED_PRINTABLE` encodings are described in RFC 2045 (MIME, Part One). The `HEXADECIMAL` encoding is a simple hexadecimal encoding of the data. The data is encoded in 8 bit byte order. Each 8 bit byte is represented with two characters; the first character describes the high four bits and the second describes the low four bits. The uuencode encoding is compatible with the popular `uuencode` and `uudecode` utilities.

`BASE64` is usually the best encoding to use: it is most likely to survive any mangling that might occur as the mail message works its way through the networks to you (*e.g.,* line wrapping, character set translation, space stripping, *etc.*).

EXAMPLES

The commands,

```
ENCODING BASE64
MODE BLOCK
SEND gif/boats*.gif
SEND/MODE=TEXT gif/index.txt
```

set the default encoding to BASE64 and the default file reading mode to BLOCK. Any files matching the specification gif/boats*.gif will be sent using these defaults. However, the file gif/index.txt will be sent as an ordinary text file owing to the use of the /MODE=TEXT qualifier.

ERROR MESSAGES

```
%MAILSERV-W-INSFPRM, missing command parameters
```

You failed to supply the name of the encoding to use. Resend the command with a valid encoding name specified.

```
%MAILSERV-W-IVKEYW, unrecognized keyword - check validity and spelling
```

You specified an unknown encoding. Resend the command with a valid encoding name specified.

END

Terminates command processing.

SYNTAX

END

DESCRIPTION

The `END` command and its synonyms `EXIT`, `FINISH`,`QUIT`, and `STOP` all cause `MAILSERV` command processing to be terminated. The remainder of the message is discarded without any additional processing.

HELP

Obtain help on using the mail server.

SYNTAX

HELP

DESCRIPTION

The `HELP` command returns a description of the commands recognized by the mail server.

ERROR MESSAGES

```
%MAILSERV-F-HLPNOTAVA, Help for server is presently unavailable
```

No help information is currently available. This may or may not be a temporary condition.

```
%MAILSERV-W-MAXPARM, too many parameters
```

You supplied a parameter after the `HELP` command. The `HELP` command does not accept any parameters (*e.g.*, does not take a “topic” parameter).

INDEX

Obtain an index of the available files.

SYNTAX

```
INDEX
```

DESCRIPTION

The INDEX command returns an index describing the files that the mail server can provide with the SEND command. This description may not give the names of each and every available file; for such information use the DIRECTORY command. The index is, typically, a simple description of some of the available files and, perhaps, a description of each of the top-level directories.

ERROR MESSAGES

```
%MAILSERV-F-INDNOTAVA, Index for server is presently unavailable
```

No file index information is currently available. This may or may not be a temporary condition. Try using the DIRECTORY command in the meantime.

```
%MAILSERV-W-MAXPARM, too many parameters
```

You supplied a parameter after the INDEX command. The INDEX command does not accept any parameters. Resend the command without any parameters.

LISTS

Obtain an index of the available mailing lists.

SYNTAX

LISTS

DESCRIPTION

The LISTS command returns an index describing the mailing lists that the mail server handles. This description may not give the names of each and every available mailing list; for such information use the DIRECTORY/LIST command. The index is, more often than not, a simple description of the mailing lists handled by the server. It may also describe any policies associated with the lists (*e.g.*, who may subscribe, how to post to the list, *etc.*).

ERROR MESSAGES

```
%MAILSERV-F-LSTNOTAVA, Index of lists is presently unavailable
```

No mailing list index information is currently available. This may or may not be a temporary condition. Try using the DIRECTORY/LIST command in the meantime.

```
%MAILSERV-F-NOMAILLIST, mailing lists are not enabled
```

The mail server is not configured to operate as a list server.

```
%MAILSERV-W-MAXPARM, too many parameters
```

You supplied a parameter after the LISTS command. The LISTS command does not accept any parameters. Resend the command without any parameters.

MAXIMUM

Set the maximum message size; larger messages will be split into several smaller messages.

SYNTAX

MAXIMUM *size-units size-value*

PARAMETERS

size-units
Required parameter specifying the units in which the **size-value** is expressed. The possible units are BYTES, BLOCKS, and LINES.

size-value
Required parameter specifying the limiting value. This must be an integer value which exceeds zero.

DESCRIPTION

Many gateways impose a limit on the maximum size message they will process. Because the mail server is often called upon to transmit large files it frequently can run afoul of such limitations.

The MAXIMUM command provides a way around such limitations. When a maximum size is set, messages larger than that size will be fragmented (split) into multiple messages, each message no larger than the specified maximum size. The fragmentation scheme is compliant with the message/partial type described in RFC 2046 (MIME, Part Two).[2]

The possible values for **size-units** are:

BYTES
size-value specifies the maximum number of bytes allowed in a single message. This value includes the initial header attached to the message. (Note that the header can increase in size through the addition of header lines during routing.)

BLOCKS
size-value specifies the maximum number of ‘‘blocks’’ of bytes allowed in a single message. The size of a block is a PMDF configuration option controlled by the system manager with the PMDF BLOCK_SIZE option; its default value is 1024 bytes. As with BYTES, this value includes the initial header attached to the message.

LINES
size-value specifies the maximum number of lines allowed in a single message. This limit is independent of the number of bytes or blocks. It is necessary to have an independent limit because some gateways limit message size based on both line count as well as overall size.

[2] PMDF systems provide facilities for automatically reassembling messages from their fragmented parts. Other mail systems that are MIME-compliant may provide similar facilities as well.

The limits specified with the MAXIMUM command apply to all subsequent SEND commands in the same message. The imposed limits may be overridden with a subsequent MAXIMUM command. And, of course, limits you imposed in previous messages sent to the server have no effect on subsequent messages which you might send.

Both line count and byte size limits may be simultaneously imposed. For instance, the two commands:

```
MAXIMUM BYTES 10000
MAXIMUM LINES 1000
```

Will result in messages larger than either 10,000 bytes or 1,000 lines being automatically fragmented into smaller messages, each containing fewer than 10,000 bytes and 1,000 lines.

See the SEND command description for further information on the usage of this command.

ERROR MESSAGES

`%MAILSERV-W-IVKEYW, unrecognized keyword - check validity and spelling`

You specified an unknown unit specification. Resend the command specifying a legal value for the **size-units** parameter.

`%MAILSERV-W-NUMBER, invalid numeric value - supply an integer`

An invalid numeric value was supplied for the **size-value** parameter. Resend the command specifying a positive integer value.

`%MAILSERV-W-POSITIVE, invalid numeric value - supply a positive integer`

An invalid numeric value was supplied for the **size-value** parameter. Resend the command specifying a positive integer value.

`%MAILSERV-W-INSFPRM, missing command parameters`

You failed to specify one or both of the required parameters. Resend the command specifying both the **size-units** and **size-value** parameters.

MODE

Set the file reading mode.

SYNTAX

MODE *mode*

PARAMETERS

mode
Required parameter specifying the file reading mode in which files are to be accessed. There are two supported modes: `TEXT` and `BLOCK`.

DESCRIPTION

Files can be read (accessed) in a variety of ways. The `MODE` command controls the method used to read the files the mail server returns. Note that default modes apply automatically to various sorts of files; this command provides a way to override these defaults.

The possible values for **mode** are:

TEXT	Read files as ordinary text files. In `TEXT` mode, files are read as a sequence of records and sent as ordinary text. `TEXT` mode is the default for files when no other mode has been set.
BLOCK	Read files as raw binary data. Any record boundary information, including carriage returns, line feeds, line length counts, and indexing information for indexed files simply becomes part of the data. The resulting data typically can only be used on the computer system it is intended for. (Note that this is not necessarily restricted to UNIX; it is possible to store files intended for other systems as UNIX files.) This is the recommend mode to use for binary files.

The reading mode specified with the `MODE` command applies to all subsequent SEND commands in the same message. It may be overridden with a subsequent `MODE` command, or, on a per command basis, with the `SEND` command's `/MODE` qualifier. And, of course, reading modes established in previous messages sent to the server have no effect on subsequent messages which you might send.

See the `SEND` command description for further information on the usage of this command.

EXAMPLES

The commands,

```
MODE BLOCK
ENCODING BASE64
SEND gif/boats*.gif
SEND/MODE=TEXT gif/index.txt
```

set the default reading mode to `BLOCK` and the default file encoding to `BASE64`. Any files matching the specification `gif/boats*.gif` will be sent using these defaults. However, the file `gif/index.txt` will be sent as an ordinary text file owing to the use of the `/MODE=TEXT` qualifier.

ERROR MESSAGES

```
%MAILSERV-W-IVKEYW, unrecognized keyword - check validity and spelling
```

You specified an unknown mode. Resend the command specifying a legal value for the **mode** parameter.

```
%MAILSERV-W-INSFPRM, missing command parameters
```

You failed to specify the **mode** parameter. Resend the command specifying a legal value for the **mode** parameter.

PURGE/LIST

Remove comment lines from a mailing list file.

SYNTAX

PURGE/LIST *list-name*

PARAMETERS

list-name
Required parameter specifying the name of the list from which comment lines are to be removed. Wildcards are not allowed.

DESCRIPTION

Mailing list files may contain comment lines. In particular, unsubscribed addresses are normally indicated via comment lines in the file. The `PURGE/LIST` command causes such comment lines to be removed, which may be useful to “clean up” the mailing list file for a list which has undergone a great many changes in membership.

EXAMPLES

The commands,

```
PURGE/LIST fads-list
SEND/LIST fads-list
```

causes the `fads-list` mailing list membership file to have comment lines removed from the file, and then a copy of the file is requested.

ERROR MESSAGES

`%MAILSERV-W-CANTUPDATE, cannot update mailing list file`

An error occurred while trying to update the mailing list file. Try again later; the postmaster in charge of the mail server has been notified.

`%MAILSERV-W-FLK, file currently locked by another user`

The specified mailing list file is not currently accessible. Try again later.

`%MAILSERV-W-INSFPRM, missing command parameters`

You failed to specify the **list-name** parameter. Resend the command specifying a legal value for the **list-name** parameter.

`%MAILSERV-W-LNF, mailing list not found`

The mailing list you specified does not exist. Resend the command specifying the name of a valid mailing list. You may use the `DIRECTORY/LIST` command to obtain a listing of the valid mailing list names.

`%MAILSERV-W-LSTCREERR, unable to create new mailing list`

The mailing list specified by the **list-name** parameter does not exist and could not be created. Check to make sure that you specified the correct list name.

`%MAILSERV-F-NOMAILLIST, Mailing lists are not enabled`

The mail server is not configured to operate as a list server.

`%MAILSERV-W-PRV, insufficient privilege or file protection violation`

You are not allowed to purge this mailing list. The `MAILSERV_ACCESS` mapping command can be used to change the default behavior of the `MAILSERV PURGE/LIST` command. Please refer to the Mail and list server section of the *PMDF System Manager's Guide*.

SEND

Retrieve one or more files from the server.

SYNTAX

SEND *file-spec extension*

Qualifiers	Defaults
/ENCODING=encoding	*None*
/MODE=mode	*/MODE=TEXT*

PARAMETERS

name
Required parameter specifying the file or files to send. This parameter can include a directory specification, but *must* include a file name. The usual shell wildcard characters, `*` and `?`, are supported.

extension
Optional parameter which can be used to specify the extension of the file to be sent.

DESCRIPTION

The SEND command sends the requested files back to you via electronic mail. Wildcards may be used in the **file-spec** parameter to specify multiple files. Each file is sent as a separate message.

The optional **extension** parameter is supplied for compatibility with BITNET's LISTSERV file servers. When supplied, a period followed by the value of this parameter will be appended to the value of the **file-spec** parameter to form the actual file name to use. For instance, the command

```
SEND newtags descript
```

is interpreted as a request for the file `newtags.descript` and is equivalent to the command

```
SEND newtags.descript
```

Large files may automatically be split into multiple smaller files prior to transmission; see the description of the MAXIMUM command for specific details. When the MAXIMUM command is used, it must be specified prior to the SEND command; *e.g.*,

```
MAXIMUM BYTES 10000
MAXIMUM LINES 1000
SEND /book/chapter*.txt
```

Files may be read in a variety of ways; this can be controlled with the MODE command or the /MODE qualifier. Files containing non-text information must be encoded in some way; the ENCODING command or the /ENCODING qualifier control the encoding used. When using the MODE and ENCODING commands, be sure to specify them before the SEND command requiring their use.

Use the DIRECTORY and INDEX commands to obtain information on available files which may be obtained with the SEND command.

QUALIFIERS

/ENCODING=encoding

The /ENCODING qualifier specifies the encoding to use for this particular file. It does not establish any default for future SEND commands, but it overrides any default set with the ENCODING command for this particular SEND command. The value is required and must be one of the values the ENCODING command accepts.

/MODE=mode

The /MODE qualifier specifies the mode to use for this particular file. It does not establish any sort of default for future SEND commands, but it overrides any default set with the MODE command for this particular SEND command. The value is required and must be one of the values the MODE command accepts.

EXAMPLES

1

```
% mail mailserv@example.com
SEND fonts/README.TXT
CTRL/D
%
```

In this example, a simple request with a single command is sent to the mail server mailserv@example.com. This single command requests that the file fonts/README.TXT be sent.

2

```
% mail mailserv@example.com
MAXIMUM BYTES 10240
SEND/MODE=BLOCK/ENCODING=BASE64 fonts/adobe35.pfb.Z
CTRL/D
%
```

In this example, a large binary file is being requested. The /MODE and /ENCODING qualifiers are used to request that the file be interpreted as raw binary data and sent in an encoded format. The MAXIMUM command is used to fragment the encoded file into several small messages, each no larger than 10K (10,240 bytes).

ERROR MESSAGES

```
%MAILSERV-W-FLK, file currently locked by another user
```

One or more of the requested files is not currently accessible. Try again later.

```
%MAILSERV-W-INSFPRM, missing command parameters
```

You failed to supply the name of the files to send. You must supply a file specification. Resend the command with a file specification.

```
%MAILSERV-W-IVKEYW, unrecognized keyword - check validity and spelling
```

You specified an unknown encoding or reading mode. Resend the command using a legal encoding or reading mode with the /ENCODING or /MODE qualifier.

```
%MAILSERV-W-NOFILES, no files found
```

Supplied file specification does not match any available files. Use the DIRECTORY command to obtain a listing of the available files.

```
%MAILSERV-F-NOFILESERV, file service is not enabled
```

The mail server is not configured to operate as a file server.

```
%MAILSERV-W-PRV, insufficient privilege or file protection violation
```

You are not allowed access to one or more of the requested files.

```
%MAILSERV-W-VALREQ, missing qualifier or keyword value
```

You failed to supply a value with the /ENCODING or /MODE qualifier. Resend the command with a value specification.

SEND/LIST

Return a list of the current subscribers to a particular mailing list.

SYNTAX

SEND/LIST *list-name*

Qualifiers	Defaults
/COMMENTS	*/COMMENTS*

PARAMETERS

list-name
Required parameter specifying the name of the list whose subscribers are to be returned. Wildcards are not allowed.

DESCRIPTION

The SEND/LIST command responds with a message containing a list of the current subscribers to a given mailing list.

QUALIFIERS

/COMMENTS
/NOCOMMENTS
When /COMMENTS is specified, comment fields associated with each subscribed address will also be returned. Specify /NOCOMMENTS to have these fields stripped from the listing sent to you. The default behavior may vary from list to list. Generally the default behavior is to include the comments.

Note that in RFC 822 addresses, comments are completely superfluous and it should be possible to strip any or all comments from an address without breaking the address. However, there are known to be mailers that incorrectly put critical information into comment fields with the expectation that the comments will not be stripped or altered. Addresses for such mailers may be rendered unreplyable by removing the comment fields from them.

ERROR MESSAGES

```
%MAILSERV-W-INSFPRM, missing command parameters
```

You failed to specify the mailing list name. Resend the command specifying the name of the mailing list whose membership list you want to obtain.

`%MAILSERV-W-LNF, list not found`

The mailing list you specified does not exist. Resend the command specifying the name of a valid mailing list. You may use the DIRECTORY/LIST command to obtain a listing of the valid mailing list names.

`%MAILSERV-F-NOMAILLIST, mailing lists are not enabled`

The mail server is not configured to operate as a list server.

`%MAILSERV-W-PRV, insufficient privilege or file protection violation`

You are not allowed to retrieve the list of subscribers to this mailing list. The MAILSERV_ACCESS mapping can be used to change the default behavior of the MAILSERV SEND/LIST command. Please refer to the Mail and list server chapter in the *PMDF System Manager's Guide*.

`%MAILSERV-W-WRITEERR, file writing error`

An error occurred while the mail server was writing the message to you. Try resending this command at a later time.

SUBSCRIBE

Subscribe to a mailing list.

SYNTAX

SUBSCRIBE *list-name [[personal-name] address]*

PARAMETERS

list-name
Required parameter specifying the name of the mailing list to subscribe to. Wildcards are not allowed.

personal-name
Optional parameter specifying the personal name for the address to subscribe to the mailing list. If this parameter is omitted, no personal name information will be included in the subscribed address.

address
Optional parameter specifying the fully-qualified address to subscribe to the mailing list. If no address is specified, the `From:` address from the requesting message will be used.

DESCRIPTION

The `SUBSCRIBE` command adds either your address or a specified address to the specified mailing list. A response message reporting the success or failure of the subscription request will be returned. If the file **list-name**`.txt` exists in the `PMDF_MAILSERV_MAIL_DIR` directory, it will be sent to you.

Use the `UNSUBSCRIBE` command to subsequently unsubscribe from a mailing list; use the `DIRECTORY/LIST` or `LISTS` command to obtain information on available mailing lists.

Note that some mail servers may impose restrictions as to who may or may not subscribe to a given list.

EXAMPLES

1
```
SUBSCRIBE local-news
```

This example shows the command to SUBSCRIBE oneself to the list local-news.

```
2   SUBSCRIBE local-news "John Doe" <jdoe+local-news@example.com>
```

This example shows the user jdoe@example.com subscribing the address "John Doe" <jdoe+local-news@example.com> to the list local-news. That is, this example shows a subscription request using a more formal address format, one that includes an RFC 822 personal name as well as the actual address, and where the address includes a subaddress; see Section 2.5 for more details about subaddresses.

ERROR MESSAGES

`%MAILSERV-W-ALREADYSUB, address is already subscribed to the mailing list`

You are already subscribed to the mailing list. If you used the optional **address** parameter, then the specified address is already subscribed. Check to make sure that you specified the correct mailing list name or address or both.

`%MAILSERV-W-CANTDELETE, cannot delete old mailing list file`

An error occurred while trying to delete the old mailing list file. Try again later; the postmaster in charge of the mail server has been notified.

`%MAILSERV-W-CANTUPDATE, cannot update mailing list file`

An error occurred while trying to update the mailing list. Try again later; the postmaster in charge of the mail server has been notified.

`%MAILSERV-W-ILLADDRESS, illegal address`

You specified an illegal or invalid address for the optional **address** parameter. Resend the command either omitting the address entirely or specifying a valid address.

`%MAILSERV-W-INSFPRM, missing command parameters`

You failed to supply the name of the mailing list to subscribe to. Resend the command with a list name specification.

`%MAILSERV-W-LNF, list not found`

The mailing list you specified does not exist. Resend the command specifying the name of a valid mailing list. You may use the `DIRECTORY/LIST` command to obtain a listing of the valid mailing list names.

`%MAILSERV-W-LSTCREERR, unable to create new mailing list`

The mailing list specified by the **list-name** parameter does not exist and could not be created. Check to make sure that you specified the correct list name.

`%MAILSERV-W-LSTLOCKED, mailing list currently locked by another user`

The mailing list is currently locked; you may not subscribe to it at this time. Try resending the command again later.

`%MAILSERV-F-NOMAILLIST, mailing lists are not enabled`

The mail server is not configured to operate as a list server.

```
%MAILSERV-W-PRV, insufficient privilege or file protection violation
```

You are not allowed to subscribe to this mailing list.

UNSUBSCRIBE

Unsubscribe from a mailing list.

SYNTAX

UNSUBSCRIBE *list-name [address]*

PARAMETERS

list-name
Required parameter specifying the name of the mailing list to unsubscribe from Wildcards are not allowed.

address
Optional parameter specifying the address to remove from the mailing list. If no address is specified, the `From:` address from the requesting message will be used.

DESCRIPTION

The `UNSUBSCRIBE` command removes either your address or the address you specify from the specified mailing list. A response message reporting the success or failure of the unsubscribe request will be returned.

Typically, the use of the optional **address** parameter is restricted.

ERROR MESSAGES

```
%MAILSERV-W-CANTDELETE, cannot delete old mailing list file
```

An error occurred while trying to delete the old mailing list file. Try again later; the postmaster in charge of the mail server has been notified.

```
%MAILSERV-W-CANTUPDATE, cannot update mailing list file
```

An error occurred while trying to update the mailing list. Try again later; the postmaster in charge of the mail server has been notified.

```
%MAILSERV-W-ILLADDRESS, illegal address
```

You specified an illegal or invalid address for the optional **address** parameter. Resend the command either omitting the address entirely or specifying a valid address.

```
%MAILSERV-W-INSFPRM, missing command parameters
```

You failed to supply the name of the mailing list to unsubscribe from. Resend the command with a list name specification.

`MAILSERV-W-LNF, mailing list not found`

The mailing list you specified does not exist. Resend the command specifying the name of a valid mailing list. You may use the `DIRECTORY/LIST` command to obtain a listing of the valid mailing list names.

`%MAILSERV-W-LSTCREERR, unable to create new mailing list`

The mailing list specified by the **list-name** parameter does not exist and could not be created. Check to make sure that you specified the correct list name.

`%MAILSERV-W-LSTLOCKED, mailing list currently locked by another user`

The mailing list is currently locked; you may not unsubscribe from it at this time. Try again later.

`%MAILSERV-F-NOMAILLIST, mailing lists are not enabled`

The mail server is not configured to operate as a list server.

`%MAILSERV-W-NOSUCHADR, no such address subscribed to the mailing list`

You are not subscribed to the specified mailing list. If you used the optional **address** parameter, then the specified address is not subscribed. Check to make sure that you specified the correct mailing list name or address or both.

`%MAILSERV-W-PRV, insufficient privilege or file protection violation`

You are not allowed to unsubscribe from this mailing list or unsubscribe addresses other than your own from the list.

6 Utilities

PMDF contains a modest collection of user-level utility programs.

Utility	Description
`db`	Manage a personal alias database
`decode`	Decode a file encoded using MIME encodings
`encode`	Encode a file using MIME encodings
`migrate`	Copy folders of messages from one system running an IMAP server to another system running an IMAP server.
`password`	Set a password in the PMDF password database
`profile`	Set delivery method in the PMDF profile database
`qm`	Manipulate your messages which you have sent but which have not yet been delivered
`send`	Send a mail message

6.1 Shell Utilities

In this section the PMDF utilities which are available as shell commands are described and examples of their usage given.

decode—Decode encoded files

Decodes a file which was previously encoded with the `pmdf encode` utility or encoded using a MIME aware mail agent.

restrictions *None.*

SYNTAX **pmdf decode** *encoded-file-spec output-file-spec*

Qualifiers	Defaults
-encoding=type	*-encoding=base64*
-filename	*-nofilename*
-header	*-header*

prompts

Input file: *encoded-file-spec*
Output file: *output-file-spec*

PARAMETERS

encoded-file-spec
Specifies the name of an encoded input file. The input file must be a file previously encoded, for example with the `pmdf encode` utility.

output-file-spec
The name of the file to produce as output. The file output by `pmdf decode` will have the identical format, structure, contents, *etc.* of the original file encoded with `pmdf encode`.

When the `-filename` qualifier is used, the output-file-spec is treated as a default file specification and as much as possible of the file name, if any, specified in the Content-type: header line is used to generate the actual output file name.

DESCRIPTION

`pmdf decode` and `pmdf encode` have been, for the most part, made obsolete by MIME-aware mail user agents, *e.g.*, Pine. If you use Pine or another MIME-aware mail user agent, then files which you send with the `Send` command will be encoded automatically, if necessary. Encoded messages which you receive will be decoded automatically, if necessary, and may simply be extracted to a file with the regular extract command. If, however, you do not use pine or another MIME aware mail user agent, then read on.

The pmdf encode and pmdf decode utilities are provided with PMDF as a means of transmitting binary files via non-MIME aware agents. With pmdf encode, a file may be encoded in a format which uses short records containing only printable characters. Such files can then be transmitted through most any mail system without being altered (*e.g.*, lines wrapped, characters removed or replaced, *etc.*). pmdf encode preserves all file contents and all file attributes when encoding a file. The contents and attributes are properly restored when decoded with pmdf decode. Absolutely any type of binary file can be transmitted with these two utilities.

Encoded files have two parts. The first part is a conventional RFC 822 message header. Header lines are used to describe the file format; this information includes a description of the encoding used to convert the file into a printable form for transfer. pmdf encode creates this header; pmdf decode reads it and uses the information it contains to reconstruct the file.

Note: Many encoded messages received with PMDF are decoded automatically for you, thus obviating the need to use pmdf decode at all. This is especially true when you use pine whose Save command will extract any MIME-encoded message or message body part. If you use mail, however, you may occasionally receive an encoded message which PMDF could not deliver in its decoded form owing to limitations of mail itself.

QUALIFIERS

-encoding=type

This qualifier controls the type of decoding used to decode the input file. The possible values for this qualifier are BASE64, CBASE64 (gzip compressed BASE64), BASE85, BINHEX (encoding only, not the file format), BTOA, HEXADECIMAL, QUOTED_PRINTABLE, UUENCODE, CUUENCODE (gzip compressed UUENCODE). It should not be necessary to specify the encoding used; this should be given in the message header. This qualifier will override the header specification if it is used.

-filename
-nofilename (default)

When the -filename qualifier is used, the output-file-spec is treated as a default file specification and as much as possible of the file name, if any, specified in the Content-type: header line is used to generate the actual output file name. The default is -nofilename in which case any file name specified in the Content-type: header line is ignored.

-header (default)
-noheader

This qualifier controls whether or not the encoded file begins with a MIME-compliant header. -header is the default. -noheader is used to force PMDF to read material that was not encoded by a MIME messaging system. When -noheader is used the -encoding qualifier is usually needed to specify the encoding since it cannot be determined from the header.

EXAMPLES

The following example illustrates a typical scenario: sue@college.edu wants to send an executable program to bob@example.com. To do this, Sue might issue the following two commands:

```
% pmdf encode program program.txt
% pmdf send -subject=\"Bob, here\'s the program\" program.txt bob@example.com
```

When Bob receives this mail message he should issue the following commands:

```
% mail
...extract the message...
% ! Remove any extra material at the beginning and ending of the file.
% pmdf decode program.txt program
```

After decoding the file, Bob may now proceed to run `program`.

Note that Sue could also have used Pine to send the file, bypassing the need to use `pmdf encode` in the first place. See the context sensitive help within Pine for help on sending files from Pine.

encode— Encode binary files

Encodes a binary file into a printable format for transmission as an e-mail message. Encoded files may be decoded with the `pmdf decode` utility. Both the standard MIME encodings as well as a few additional encodings (*e.g.,* UUENCODE) are supported.

restrictions *None.*

SYNTAX

pmdf encode *input-file-spec encoded-file-spec*

Qualifiers	**Defaults**
-encoding=type	*-encoding=BASE64*
-filename	*-nofilename*
-header	*-noheader*

prompts

Input file: *input-file-spec*
Output file: *encoded-file-spec*

PARAMETERS

input-file-spec

Specifies the name of an input file. The input file may be any binary file. Only a single input file may be specified; wildcards are not allowed.

encoded-file-spec

The name of the file to produce as output. The file output by `pmdf encode` will contain all of the information necessary to reconstruct the original input file. The format of the output file is described in the Description section below.

DESCRIPTION

`pmdf decode` and `pmdf encode` have been, for the most part, made obsolete by MIME-aware user agent such as pine. If you use pine, then files which you send with the `Send` command will be encoded automatically, if necessary. Encoded messages which you receive will be decoded automatically, if necessary, and may simply be extracted to a file with the `Save` command. If, however, you do not use a MIME aware user agent such as pine, then read on.

The `pmdf encode` and `pmdf decode` utilities are provided with PMDF as a means of transmitting binary files via e-mail. With `pmdf encode`, a file may be encoded in a format which uses short records containing only printable characters. Such files can then be transmitted through most any mail system without being altered (*e.g.*, lines wrapped, characters removed or replaced, *etc.*). `pmdf encode` preserves all file contents and all file attributes when encoding a file. The contents and attributes are properly restored when decoded with pmdf decode. Absolutely any type of file can be transmitted with these two utilities.

Encoded files have two parts. The first part is a conventional RFC 822 message header. Header lines are used to describe the file format; this information includes a description of the encoding used to convert the file into a printable form for transfer. pmdf encode creates this header; pmdf decode reads it and uses the information it contains to reconstruct the file.

Note: Many encoded messages received with PMDF are decoded automatically for you, thus obviating the need to use pmdf decode at all. This is especially true when you use pine whose Save command will extract any MIME-encoded message or message body part. If you use a non-MIME aware user agent, however, you may occasionally receive an encoded message which PMDF could not deliver in its decoded form to mail owing to limitations of mail itself.

QUALIFIERS

-encoding=type

This qualifier controls the type of encoding used to encode the input file. The possible values for this qualifier are BASE64, CBASE64 gzip compressed BASE64, BASE85, BINHEX (encoding only, not the file format), BTOA, HEXADECIMAL, QUOTED_PRINTABLE, UUENCODE, CUUENCODE (gzip compressed UUENCODE). BASE64 encoding is the default; this is also the default decoding type used by pmdf decode.

-filename
-nofilename (default)

When used in conjunction with the -header qualifier, this qualifier specifies that the filename should be included in the MIME headers generated. Only the name and extension portion of the input file specification will be used; will be discarded. By default, no filename parameter is specified in the Content-type: or Content-disposition: header lines.

Or if used with -encoding=uuencode, the -filename qualifier causes the filename to be included on the begin 600 line.

-header
-noheader (default)

This qualifier controls whether or not a MIME-compliant header is placed at the beginning of the output. -header is the default. -noheader is used to produce output suitable for use in non-MIME messaging applications. Note that all structural information about the file is lost when -noheader is used.

EXAMPLES

See the example provided for the pmdf decode command. In that example, the use of pmdf encode is also demonstrated.

migrate—Copy messages from one system to another system

Copy folders of messages from one system running an IMAP server to another system running an IMAP server.

SYNTAX

migrate *-fromhost source-IMAP-host -fromuser source-username -frompass source-password -tohost destination-IMAP-host -touser destination-username -topass destination-password [optional-options]*

Qualifiers	Defaults
-batch	*Move in batch mode without prompting; see text*
-debug n	*Specify level of debugging*
-freq n	*How frequently to report messages moved*
-fromport port	*Specify a non-default port number*
-fromprefix folder-prefix	*See text*
-fromrecon n	*Reconnect to source host periodically*
-h	*Print help message*
-log file-spec	*Log operations to specified file*
-lower	*Destination folder names will be lowercased*
-maxerr n	*Maximum number of errors to allow before aborting*
-noempty	*Skip empty folders*
-noinbox	*Skip INBOX on source host*
-nowastebasket	*Skip WASTEBASKET on source host*
-public	*Copy public folders*
-recover	*Attempt to recover if there is a communication failure*
-toport port	*Specify a non-default port number*
-toprefix folder-prefix	*See text*
-torecon n	*Reconnect to destination host periodically*

prompts

Enter TCP/IP name of host to move from:	*source-IMAP-host*
Enter TCP/IP name of host to move to:	*destination-IMAP-host*
{*source-IMAP-host*} username:	*source-username*
{*source-IMAP-host*} password for *username*:	*source-password*
{*destination-IMAP-host*} username:	*destination-username*
{*destination-IMAP-host*} password for *username*:	*destination-password*
Move *n* messages in *source-folder* to which folder:	*destination-folder*

PARAMETERS

source-IMAP-host
The fully qualified TCP/IP name of the system from which to copy messages. The system must be running an IMAP server.

source-username
The username of the account on the source system.

source-password
The password of the account on the source system.

destination-IMAP-host
The fully qualified TCP/IP name of the system to which to copy messages. The system must be running an IMAP server.

destination-username
The username of the account on the destination system.

destination-password
The password of the account on the destination system.

source-folder-prefix
A string specifying what folders to copy from.

destination-folder-prefix
A string specifying what folders to copy to.

DESCRIPTION

`migrate` is a utility for copying folders of messages from one system to another. Each system must have an IMAP server running.

You can run migrate interactively or in batch mode. To run it interactively, you will be prompted for the names of the systems to move from and to, as well as the usernames and passwords. If all the required arguments are specified, then migrate will run in automatic or batch mode, moving all messages from all folders to the destination system, using the same name for destination folders. With the optional `-batch` qualifier, you can skip the interactive confirmation prompt before each folder is copied. With the `-fromprefix` qualifier, you can specify a folder prefix for the source system; with the `-toprefix` qualifier, you can specify a folder prefix for the destination system. And with the `-lower` qualifier you can specify that the destination folder names should be lowercased. If you want to skip empty folders, specify `-noempty`. If you want public folders to be copied over as well, specify `-public`.

Assuming that the source system is VMS, the folders INBOX and WASTEBASKET are not skipped. If you want to skip the INBOX folder, specify `-noinbox`. If you want to skip the WASTEBASKET folder, specify `-nowastebasket`.

There can be a communications failure to either the source host or the destination host or both. By default, `migrate` exits immediately upon failure. Specify `-recover` to have migrate attempt recovery after a communications failure. Similarly, there can be problems in the IMAP servers on either the source or destination, such as the servers running out of memory if there are many folders with many large messages being migrated. Specify `-fromrecon` or `-torecon` to have migrate periodically disconnect and reconnect to the source or destination IMAP server.

QUALIFIERS

-batch
Do not prompt for confirmation of what folders to move.

-debug *n*
This qualifier enables debug output. *n* is a bit-encoded integer, with bits:

1	prints out progress/delay/login/logout messages
2	prints certain responses received, but not full message text
4	prints responses received, including full message text
8	prints commands sent

-freq *n*
Specify how freqently to report messages moved. Defaults to every 100 messages during copy.

-fromport *port*
Specify the IMAP port on the source system. The default is `-fromport 143`, the standard IMAP port.

-fromprefix *folder-prefix*
Specify the folder prefix for the source mailbox, if not the default mail file. For instance, if migrating from a PMDF IMAP server on OpenVMS system, the folder prefix would typically have the form `#`*`disk`*`:[`*`dir`*`]`*`mailfile`*`.mai#` or if migrating from an IMAP server on a UNIX system, the folder prefix would be a UNIX file specification (the UNIX default typically being `mail/`).

-fromrecon *n*
This qualifier tells migrate to disconnect and reconnect to the source host IMAP server every **n** number of messages. The default is to never disconnect and reconnect.

-h
Print help text.

-log *file-spec*
Specify a file in which to log operations.

-lower
Lowercase the folder names for the destination system.

-maxerr *n*
Specify the maximum number of errors that may be encountered before aborting. The default, if this qualifier is not specified, is to continue after all errors.

-noempty
Skip copying empty folders.

-noinbox

By default, the INBOX folder is not skipped when copying folders; the **-noinbox** qualifier instructs PMDF to skip this folder.

-nowastebasket

By default, the WASTEBASKET folder is not skipped when copying folders; the **-nowastebasket** qualifier instructs PMDF to skip this folder.

-public

Copy public folders on the source host. The default is to not copy folders with these prefixes: “Other Users/”, “Public Flders/”, “user.”. Specifying the **-public** qualifier instructs PMDF to not skip such folders.

-recover

Attempt recovery if there is a communications failure to one of the IMAP servers. The default is to not attempt recovery, but to exit instead.

***-toport** port*

Specify the IMAP port on the destination system. The default is `-toport 143`, the standard IMAP port.

***-toprefix** folder-prefix*

Specify the folder prefix for the destination mailbox, if not the default mail file. For instance, if migrating to a PMDF IMAP server on OpenVMS system, the folder prefix would typically have the form `#`*`disk`*`:[`*`dir`*`]`*`mailfile`*`.mai#` or if migrating to an IMAP server on a UNIX system, the folder prefix would be a UNIX file specification (the UNIX default typically being `mail/`).

***-torecon** n*

This qualifier tells migrate to disconnect and reconnect to the destination host IMAP server every **n** number of messages. The default is to never disconnect and reconnect.

EXAMPLES

1

```
% /pmdf/bin/migrate
From which host: admin.example.com
Username on admin.example.com: alonso
Password for alonso on admin.example.com:
To which host: hq.example.com
Username on hq.example.com: alonso
Password for alonso on hq.example.com:
Move 15 messages in sentmail to which folder (CR to skip): SENTMAIL
Move 4 messages in mail to which folder (CR to skip): RETURN
... Skipping mail
```

In this example user alonso on admin.example.com also has an account on hq.example.com with two folders, `mail` and `sentmail`, and is copying the messages in the sentmail folder on admin.example.com to the `SENTMAIL` folder on hq.example.com.

password—Set remote authentication password

Set password for remote authentication, *e.g.*, POP client (APOP), IMAP client (CRAM), or mailbox filter authentication.

SYNTAX

pmdf password *[password]*

Qualifiers	**Defaults**
`-create`	`-create`
`-delete`	`-create`
`-service=`*keyword*	`-service=DEFAULT`
`-show`	`-create`
`-test`	`-create`

restrictions

All operations other than setting, deleting, or verifying one's own password require privileges.

prompts

Password: *password*

PARAMETERS

password
The password to set.

DESCRIPTION

The `pmdf password` utility is used to add and change password values that may be used for special authorization purposes. Whether you need to use this utility will depend on your site's configuration. If you use the native Berkeley mailbox from POP or IMAP clients, then you may need to. Otherwise you probably will not; for instance, PMDF popstore users and PMDF MessageStore users typically do not need to be concerned with password database entries. See Section 7.1 for a further discussion of the `pmdf password` utility as it relates to POP and IMAP client usage. And check with your system administrators if you are uncertain about whether you need to use this utility (to set a PMDF password database entry or entries).

Typically, if you use the `pmdf password` utility at all you would use it to create just a `DEFAULT` service entry. However, it is possible to instead have service specific entries; a single user can have separate entries for separate services, such as an entry for the IMAP service (preferentially used when authenticating during IMAP connections), and an entry for the POP service (preferentially used when authenticating during POP connections). Authentication checks preferentially for an appropriate service specific entry, but if a service specific entry is not present then falls through to checking the DEFAULT service entry.

QUALIFIERS

-create
Add a new password database entry. This qualifier is the default.

-delete
Delete a user/password entry pair from the PMDF password database.

-service=*keyword*
Specify for what service a particular password method and password value apply. The default service keyword is DEFAULT; POP3 and IMAP are other possible keywords.

-show
Show a user/service/password-method entry in the PMDF password database. Note that this command does not show the password value.

-test
Compare a specified password against a password stored in the PMDF password database.

EXAMPLES

To set your password, with prompting so that the password is not displayed on the screen as you type it, issue the command:

```
# pmdf password
Password:
```

ERROR MESSAGES

`entry too long to fit in password file`

The password entry is too long; try using a shorter password value.

`cannot open password file`

The PMDF password database does not exist, or could not be opened. Check with your system administrator.

send— Send a mail message

Sends a mail message using PMDF.

restrictions *None.*

SYNTAX **pmdf send** *message-file-spec[,...] recipient-address[,...]*

Qualifiers	**Defaults**
-abort	*See text*
-comments=comment	*None*
-delivery_receipt_to=address	*None*
-errors_to=address	*None*
-expand_limit=limit	*None*
-extra=header_line	*None*
-from=address	*See text*
-headers	*-noheaders*
-ignore	*See text*
-importance=importance	*None*
-keywords=keywords	*None*
-log=log-list	*-nolog*
-multipart	*None*
-organization=organization	*None*
-priority=priority	*None*
-read_receipt_to=address	*None*
-references=references	*None*
-reply_to=address	*None*
-return_address=address	*See text*
-rfrom=address	*None*
-rreply_to=address	*None*
-sensitivity=sensitivity	*None*
-subaddress=subaddress	*None*
-subject=subject	*None*
-user=username	*See text*
-warnings_to=address	*None*
-x_ps_qualifiers=qualifiers	*None*

Positional Qualifiers	**Defaults**
-bcc	*-to*
-cc	*-to*
-encoding=encoding	*See text*
-filename=name	*-nofilename*
-mode=mode	*-mode=TEXT*
-to	*-to*

prompts Message file: *message-file-spec[,...]*

Address: *recipient-address*

PARAMETERS

message-file-spec[,...]
One or more files to comprise the message; wildcards are not allowed. Each file is included in the mail message as a separate part.

recipient-address[,...]
The recipients who are to receive copies of the message. Standard RFC 822 format addresses must be used. Quoting may be needed to preserve case and special characters.

DESCRIPTION

The `pmdf send` utility provides a simple easy-to-use interface to PMDF for sending messages.

QUALIFIERS

-abort
By default, if an error occurs while processing an input message file or recipient address, `pmdf send` will ask the user whether or not to send the message anyhow. If the `-abort` qualifier is specified, then `pmdf send` will merely exit (with an error) when a problem occurs during file or address processing.

The `-abort` and `-ignore` qualifiers are mutually exclusive — only one or the other may be used.

-bcc
Positional parameter which may be used to specify that the given recipient address, and subsequent recipient addresses, should be treated as `Bcc:` addresses. By default, recipient addresses are interpreted as `To:` addresses. Must be placed after the address to which it should apply.

-cc
Positional parameter which may be used to specify that the given recipient address, and subsequent recipient addresses, should be treated as `cc:` addresses. By default, recipient addresses are interpreted as `To:` addresses. Must be placed after the address to which it should apply.

-comments=comments
This qualifier may be used to specify the contents of the `Comments:` header line. If this qualifier is not used, any existing `Comments:` header line is used; if none exists no `Comments:` header line will appear in the outgoing message unless the `PMDF_COMMENTS` environment variable is defined.

-delivery_receipt_to=address
This qualifier may be used to specify the contents of the `Delivery-receipt-to:` header line. If this qualifier is not used, any existing `Delivery-receipt-to:`

header line is used; if none exists, no Delivery-receipt-to: header line will appear in the outgoing message.

-encoding=encoding

Specify the encoding method to use to encode an input message file. Normally, no encoding is used; however, this depends upon the file type as determined by the file extension. The available encoding methods are BASE32, BASE64, CBASE64 (compressed base64), BASE85, BINHEX, BTOA HEXADECIMAL, QUOTED_PRINTABLE, UUENCODE, and CUUENCODE (compressed UUENCODE). No encoding may be specified for a file containing header information (-header). Must be placed after the address to which it should apply.

-errors_to=address

This qualifier may be used to specify the contents of the Errors-to: header line. If this qualifier is not used, any existing Errors-to: header line is used; if none exists no Errors-to: header line will appear in the outgoing message unless the PMDF_ERRORS_TO environment variable is defined.

-expand_limit=limit

If, during the process of expanding the message's recipient addresses, the count of recipients exceeds the specified limit then the address expansion will be deferred. pmdf send will expand the addresses "off-line" so that the user need not wait.

-extra=header_line

Additional header lines may be specified with the -extra qualifier. Specify the entire text of the header line; *e.g.*, -extra="X-Sign: Aquarius". Multiple header lines should be specified using the format -extra=(hdr1,hdr2,...); note that you will need to escape the parentheses characters to the shell by quoting each with a backslash.

-filename=name
-nofilename (default)

This positional qualifier, when specified, causes the name of the input file to be included as a parameter to the associated MIME Content-type: header line. Must be placed after the input file to which it should apply. You can specify the name to be included in the header line, or if you do not specify a name, by default the name of the input file will be used.

-from=address

This qualifier may be used to specify the contents of the From: header line. If this qualifier is not used, any existing From: header line is used; if none exists and the PMDF_FROM environment variable is not defined, then a From: header line will be constructed from the username of the user invoking pmdf send and from the local host name. Note that even if a From: address is provided your address will appear in a Sender: header line.

-headers
-noheaders (default)

The input message is assumed to have no header attached to it by default. The -headers qualifier tells pmdf send that a header is already attached to the message; it is modified and used to form the header for the message that is actually sent.

-ignore

By default, if an error occurs while processing an input message file or recipient address, pmdf send will ask the user whether or not to send the message anyhow. If the -ignore qualifier is specified, then pmdf send will not ask the user whether or not to send the message — it will send the good input files to the good recipient addresses.

The -abort and -ignore qualifiers are mutually exclusive — only one or the other may be used.

-importance=importance

This qualifier may be used to specify the contents of the Importance: header line. If this qualifier is not used, any existing Importance: header line is used; if none exists no Importance: header line will appear in the outgoing message unless the PMDF_IMPORTANCE environment variable is defined.

-keywords=keywords

This qualifier may be used to specify the contents of the Keywords: header line. If this qualifier is not used, any existing Keywords: header line is used; if none exists no Keywords: header line will appear in the outgoing message unless the PMDF_KEYWORDS environment variable is defined.

-log=log-list

Specify what sort of informational message pmdf send should issue. The log-list is a list of zero or more of the following: NONE, ADDRESSES, FILES, MESSAGES, IDS, or ALL. NONE indicates no logging, is equivalent to -nolog, and is the default. ADDRESSES causes one informational message to be output for each specified recipient address. FILES causes one information message to be output for each input file. MESSAGES produces a summary message that indicates how many addresses and files were processed successfully. IDS produces an informational message showing the contents of the Message-Id: header of the resulting message. ALL activates all forms of logging; it cannot be specified simultaneously with NONE. -log=messages is the default if -log is specified without an explicit log-list.

-mode=mode

Specify the file access mode to use when reading an input message file. By default, input files are read in text mode. The access modes are TEXT and BLOCK. A file containing header information must be accessed using TEXT mode. Must be placed after the address to which it should apply.

-multipart

This qualifier may be used to tell pmdf send to always format the messages it sends as multipart MIME messages.

-organization=organization

This qualifier may be used to specify the contents of the Organization: header line. This qualifier is ignored if an Organization: header line is already present. If this qualifier is not specified or negated, no Organization: header line is added unless the PMDF_ORGANIZATION environment variable is defined.

-priority=priority

This qualifier may be used to specify the contents of the Priority: header line. If this qualifier is not used, any existing Priority: header line is used; if none exists, no Priority: header line will appear in the outgoing message.

-read_receipt_to=address

This qualifier may be used to specify the contents of the `Read-receipt-to:` header line. If this qualifier is not used, any existing `Read-receipt-to:` header line is used; if none exists, no `Read-receipt-to:` header line will appear in the outgoing message.

-references=references

This qualifier may be used to specify the contents of the `References:` header line. If this qualifier is not used, any existing `References:` header line is used; if none exists no `References:` header line will appear in the outgoing message unless the `PMDF_REFERENCES` environment variable is defined.

-reply_to=address

This qualifier may be used to specify the contents of the `Reply-to:` header line. If this qualifier is not used, any existing `Reply-to:` header line is used; if none exists, no `Reply-to:` header line will appear in the outgoing message unless the `PMDF_REPLY_TO` environment variable is defined.

-return_address=address

Specify the address to be used as the envelope originator address. If the message is returned as undeliverable by the mail system transport the non-delivery notice is normally sent to this address.

-rfrom=address

This qualifier may be used to specify the contents of the `Resent-From:` header line. If this qualifier is not used, any existing `Resent-From:` header line is used; if no such header line exists, no `Resent-From:` header line will be attached to the outgoing message.

-rreply_to=address

This qualifier may be used to specify the contents of the `Resent-Reply-to:` header line. If this qualifier is not used, any existing `Resent-Reply-to:` header line is used; if no such header line exists no `Resent-Reply-to:` header line will be attached to the outgoing message.

-sensitivity=sensitivity

This qualifier may be used to specify the contents of the `Sensitivity:` header line. If this qualifier is not used, any existing `Sensitivity:` header line is used; if none exists, no `Sensitivity:` header line will appear in the outgoing message unless the `PMDF_SENSITIVITY` environment variable is defined.

-subaddress=subaddress

Specify a subaddress to attach to the envelope `From:` address; *e.g.*, if the envelope `From:` address is rex@example.com then specifying `-subaddress="Postmaster"` would result in the envelope `From:` address rex+Postmaster@example.com.

-subject=subject

This qualifier may be used to specify the contents of the `Subject:` header line. If this qualifier is not used, any existing `Subject:` header line is used; if none exists no `Subject:` header line will appear in the outgoing message.

-to

Positional parameter which may be used to specify that a given recipient address should be treated as a `To:` address, which is the default interpretation. Must be placed after the address to which it should apply.

-user=username

Specify the local username to use in the message sender's address; (this will be the From: address if no other `From:` address is given and the `Sender:` address otherwise). You must either have superuser privilege or be in the `pmdf_world` or `pmdf_world_`*`username`* group id in order to use this qualifier and specify a username other than your own. The special case of a blank string will not insert any `Sender:` information.

-warnings_to=address

This qualifier may be used to specify the contents of the `Warnings-to:` header line. If this qualifier is not used, any existing `Warnings-to:` header line is used; if none exists, no `Warnings-to:` header line will appear in the outgoing message unless the `PMDF_WARNINGS_TO` environment variable is defined.

-x_ps_qualifiers=qualifiers

This qualifier may be used to specify the contents of the `X-ps-qualifiers:` header line. If this qualifier is not used, any existing `X-ps-qualifiers:` header line is used; if none exists no `X-ps-qualifiers:` header line will appear in the outgoing message unless the `PMDF_X_PS_QUALIFIERS` environment variable is defined.

EXAMPLES

1

```
% pmdf send -subject=\"Test message\" msg.txt bob@example.com
```

This command will send as a mail message the contents of the file `msg.txt` to the address bob@example.com. The `Subject:` line of the message will read `Subject: Test message`.

2

```
% pmdf send -subject=\"Test message\" \
      -extra='("X-Favorite-Drink: Hot chocolate", \
      "X-IQ: 20/20")' msg.txt bob@example.com
```

Send a message to bob@example.com with the header lines

```
Subject: Test message
X-Favorite-Drink: Hot chocolate
X-IQ: 20/20
```

(Of course, these will not be the only header lines present.)

3

```
% pmdf send -headers hdr.txt,msg.txt bob@example.com
```

Send a message to bob@example.com. The headers for the message will be derived from the file `hdr.txt` while the body of the message will be the contents of the file `msg.txt`.

4

```
% pmdf send msg.txt -filename bob@example.com -to, sue@college.com -cc
```

This command shows the use of the positional qualifiers `-filename`, `-to`, and `-cc`. They are placed after the parameter to which they apply.

6.2 profile: Setting a Delivery Method

The pmdf profile utility sets PMDF user profile database entries. It may be used by the PMDF system manager to define delivery methods for local users, and to set the delivery method for a specific user or to set a default delivery method for local users, and may also be used by users to select among the defined delivery methods. For details on using pmdf profile at the system level (to define delivery methods), see the *PMDF System Manager's Guide*; this section will discuss only the user level commands, for setting and displaying one's own delivery method.

To invoke the pmdf profile utility, issue the command

```
# pmdf profile
```

Use the exit or quit command to exit pmdf profile.

The available commands for users are summarized in Table 6–1.

Table 6–1 Summary of pmdf profile commands

delete delivery	Clear a delivery method selection
set delivery	Select a delivery method
show delivery	Show what delivery method is currently selected
show method	Show the definition of a delivery method

delete delivery

Clear a delivery method selection.

SYNTAX

delete delivery

Command Qualifiers	**Defaults**
None.	*None.*

restrictions

Must be `root` in order to delete another user's choice of delivery method, or to delete the default delivery method selection.

PARAMETERS *None.*

DESCRIPTION

This utility is used to delete (clear) a previous delivery method selection from the PMDF user profile database. A user may only delete their own delivery method choice.

EXAMPLES

Below is an example of deleting a delivery method selection.

```
# pmdf profile
profile> show delivery
  Your delivery selection is BSD
profile> delete delivery
profile> exit
```

set delivery

Select a delivery method.

SYNTAX

set delivery *method*

Command Qualifiers	**Defaults**
None.	*None.*

restrictions

Must be `root` in order to define a delivery method for a user other than oneself, or to set a default delivery method.

PARAMETERS

method
A delivery method previously defined for the PMDF profile database by the PMDF postmaster.

DESCRIPTION

This utility is used to select a delivery method in the PMDF user profile database. A user may only select a delivery method for themself.

EXAMPLES

Below is an example of setting one's own delivery method to DMW.

```
# pmdf profile
profile> show method -all
Method BSD is defined as: /var/spool/mail/%s
Method DMW is defined as: |/usr/bin/inetgrecv %s
Method MIME is defined as: +/var/spool/mail/%s
profile> set delivery DMW
profile> exit
```

show delivery

Show a delivery method.

SYNTAX

show delivery *NONE*

Command Qualifiers	Defaults
None.	*None.*

restrictions Must be `root` in order to show a delivery method for a user other than oneself, or to show the default delivery method, or to show the delivery method selected by each user.

PARAMETERS

None.

DESCRIPTION

This utility is used to show a delivery method in the PMDF user profile database. A user may only show their own selected delivery method.

EXAMPLES

Below is an example in which user alonso shows his own selected delivery method.

```
% pmdf profile
profile> show delivery
User alonso delivery selection is currently set to DMW
profile> exit
```

show method

Show a delivery method definition.

SYNTAX

show method *[method-name]*

Qualifiers	Defaults
`-all`	*See text*

restrictions *None.*

PARAMETERS

method-name
An optional parameter, which if specified must be the name of a previously defined delivery method.

DESCRIPTION

This command is used to show the definitions of delivery methods in the PMDF user profile database. This command may be used to show the definition of a particular delivery method, by specifying the `method` parameter, or may be used to show the definitions of all the currently defined delivery methods by instead specifying the `-all` qualifier.

COMMAND QUALIFIERS

`-all`
Show the definitions of all delivery methods.

EXAMPLES

Below is an example of showing all defined delivery methods.

```
# pmdf profile
profile> show method -all
Method BSD is defined as: /var/spool/mail/%s
Method DMW is defined as: |/usr/bin/inetgrecv %s
Method MIME is defined as: +/var/spool/mail/%s
profile> exit
```

6.3 qm: Manipulate Your Queued Messages

pmdf qm is a utility program which allows inspection and manipulation of queued messages. While pmdf qm has two modes, maintenance mode and user mode, unprivileged users may only use the user mode. From user mode, you can obtain listings of all “queued” messages which you have sent but which are still on your system awaiting delivery.[1] You can read or return any of your queued messages.

Note that this utility merely reports on messages in PMDF’s delivery queues. That a message you have sent no longer appears in PMDF’s queues, does not imply that it has reached its final destination. All that it means is that the message has left the PMDF system and is no longer under PMDF’s control. For example, it is not uncommon for a message to make an intermediate stop on another system such as a mail hub. In such cases, PMDF will consider the message to be “delivered” when it hands the message and responsibility for it off to the intermediate system.

To run pmdf qm, issue the command

```
% pmdf qm
```

Use the exit or quit command to exit pmdf qm. The commands accepted by this utility in user mode are summarized in Table 6–2 and described in detail below.

Table 6–2 Summary of pmdf qm user mode commands

date	Show current date and time
directory	List currently queued messages
exit	Exit the utility
help	Obtain help
history	Display message delivery history information
quit	Exit the utility
read	Display message envelope and header information
return	Return a message to its originator
run	Execute file of commands

The command recall and editing capabilities are provided by the open source software libedit (also known as editline). By default, the standard "vi" key bindings are defined. You can change various elements of the editing environment, such as using "Emacs" key bindings instead of "vi", by creating in your home directory a file called .editrc. See the editrc manpage for more information.

[1] Usually, when you send a message an immediate attempt is made to deliver it. Should that attempt fail owing to a temporary problem such as a network outage, the message will be queued in the mail system’s delivery queues. Subsequent, periodic attempts will be made to deliver the message until either it is delivered or it is determined to be undeliverable in which case it is returned to you. Messages are typically returned as undeliverable because the address turns out to be incorrect; *e.g.*, the destination system does not recognize the recipient address or the destination system is unreachable.

date

Show the current date and time.

SYNTAX

date

Command Qualifiers	Defaults
None.	*None.*

PARAMETERS

None.

DESCRIPTION

The `date` command may be used to show the current date and time, in RFC 822/1123 format — the same format as used in Internet-style messages.

EXAMPLES

In the following example, the current date and time in RFC 822/1123 format is displayed with the `date` command.

```
qm.user> date
Tue, 3 Jun 2012 13:34:16 -0400 (PST)
qm.user>
```

directory

List currently queued messages.

SYNTAX

directory *[type]*

Command Qualifiers	Defaults
None.	*None.*

PARAMETERS

type
An optional parameter specifying the type of messages to display (*e.g.*, Internet, cc:Mail, *etc.*). Wild cards are permitted.

DESCRIPTION

Use the `directory` command to list any messages which you have sent but which have not yet been delivered.[2] The optional **type** argument may be used to restrict the listing to certain types of messages such as messages sent to the Internet or other TCP/IP connected machines such as UNIX workstations, cc:Mail users, *etc.* You can use the command `help directory` to obtain a complete list of the available types.

Type	Message types listed
ccmail	Messages sent to Lotus cc:Mail users
groupwise	Messages to GroupWise Office users
internet	Messages sent to Internet users
local	Messages sent to local BSD mailbox users
lotus_notes	Messages sent to Lotus Notes users
mailbus_400	Messages sent to MAILbus 400 users
microsoft_mail	Messages sent to Microsoft Mail users
netdata	Messages sent to Netdata (PROFS) users
novell_mhs	Messages sent to Novell MHS users
ovvm	Messages sent to OV/VM (PROFS) users
pager	Messages sent to personal pagers
popstore	Messages sent to popstore users
snads	Messages sent to SNADS users

[2] Note for system managers: unlike the `pmdf qm` utility's maintenance mode, in user mode the directory listing is always generated from the queue cache database.

Type	Message types listed
tcpip	Messages sent to TCP/IP users
uucp	Messages sent to UUCP users
wordperfect	Messages sent to WordPerfect Office users
x400	Messages sent to X.400 users

In the directory listing, each message is assigned a message identification number, or “message id” for short. The message id appears in the leftmost column. These identification numbers may be used with the `read` or `return` commands to identify which messages to read or return.

It is important to note that when you send a message to more than one recipient, the message may be split into multiple message copies. Consequently, the same message may appear multiple times as being queued to different networks (or possibly even for the same network). Such would be the case for a message sent both to local users and remote users.

EXAMPLES

```
qm.user> directory
Thu, 12 Jul 2012 18:49:40 -0800 (PST)

Id Network           From          To                  Size Queued since
---------------------------------------------------------------------------
 1 Internet (TCP/IP) bob@example.com  service@example.com    8 12-MAR 17:31
                                      service@internode.co
 2 Internet (TCP/IP) bob@example.com  ietf-822@dimacs.rut    8 12-MAR 15:07
 3 Internet (TCP/IP) bob@example.com  mwalnut@cnri.reston   16 12-MAR 15:26
 4 Internet (TCP/IP) bob@example.com  jbakin@adoc.xerox.com  8 12-MAR 17:18
 5 Internet (TCP/IP) bob@example.com  klensin@MAIL1.RESTO   16 12-MAR 15:26
 6 Internet (TCP/IP) bob@example.com  MAILSERV@example.COM   8 12-MAR 15:38
 7 Internet (TCP/IP) bob@example.com  ned@example.COM      8 12-MAR 17:18
 8 Lotus Notes       bob@example.com  John.Smith@notes.ac    8 10-MAR 12:25
 9 Local delivery    bob@example.com  ned                    8 11-MAR 16:11
10 Internet (TCP/IP) bob@example.com  mailserv@example.com   8 12-MAR 12:43
11 Internet (TCP/IP) bob@example.com  MARKJOSEPH@delphi.com  8 12-MAR 15:07
---------------------------------------------------------------------------
Total size:                                                104

qm.user>
```

In this example, the `directory` command is used to list all queued messages. When a message has more than one envelope `To:` recipient, the additional recipients are shown on additional lines of the listing as with message 1 which is addressed to service@example.com and service@internode.com.au.

exit

Exit the `pmdf qm` utility.

SYNTAX

exit

Command Qualifiers	Defaults
None.	*None.*

PARAMETERS

None.

DESCRIPTION

The `exit` and `quit` commands exit the `pmdf qm` utility.

help

Obtain help on the use of `pmdf qm`.

SYNTAX

help *[topic]*

Command Qualifiers	**Defaults**
None.	*None.*

PARAMETERS

topic
Optional topic to obtain help on.

DESCRIPTION

The `help` command may be used to obtain information on `pmdf qm` commands. To obtain information on all of the `pmdf qm` commands, use the command

```
qm.user> help
```

To obtain information on individual commands or topics use the command

```
qm.user> help topic
```

where *`topic`* is the name of the command or topic of interest.

history

Display message history information.

SYNTAX

history *[message-id[,...]]*

Qualifiers	Defaults
`-all`	`-noall`
`-confirm`	`-noconfirm`

PARAMETERS

message-id
A comma separated list of one or more message identification numbers shown with a previous `directory` command. Ranges are allowed.

DESCRIPTION

For many channels, delivery history information is appended to the end of each message file after an unsuccessful delivery attempt has been made. With the `history` command, this information can be displayed.

The messages to show histories for are specified by their message identification numbers shown by the most recent `directory` command. That number appears in the leftmost column of the `directory` command listing.

Note that history information is not recorded by some channels.

QUALIFIERS

`-all`
`-noall` ***(default)***
Display history information for all messages shown with the last `directory` command.

`-confirm`
`-noconfirm` ***(default)***
When `-confirm` is specified, you will be prompted to confirm whether or not to display the history for each selected message.

quit

Exit the *pmdf qm* utility.

SYNTAX

quit

Command Qualifiers	Defaults
None.	*None.*

PARAMETERS

None.

DESCRIPTION

The `exit` and `quit` commands exit the `pmdf qm` utility.

read

Read a message.

SYNTAX

read *[message-id[,...]]*

Qualifiers	Defaults
`-all`	`-noall`
`-confirm`	`-noconfirm`
`-content`	`-content`

PARAMETERS

message-id
A comma separated list of one or more message identification numbers shown with a previous `directory` command. Ranges are allowed.

DESCRIPTION

The `read` command may be used to read one or more queued messages. The messages to display are specified by their message identification numbers shown by the most recent `directory` command. Those numbers appear in the leftmost column of the `directory` command listing.

QUALIFIERS

`-all`
`-noall` ***(default)***
Display all messages shown with the last `directory` command.

`-confirm`
`-noconfirm` ***(default)***
When `-confirm` is specified, you will be prompted to confirm whether or not to display each selected message.

`-content` ***(default)***
`-nocontent`
Specify `-nocontent` if you only want to read the message envelope and header.

EXAMPLES

In the following example, message 3 is displayed.

```
qm.user> read 3
Message id: 3

Transport layer information:
----------------------------------------------------------------------
Envelope From: address: prospero@example.com
Envelope To: addresses: alonso

Message header:
----------------------------------------------------------------------
Received: from EXAMPLE.COM by EXAMPLE.COM (PMDF V6.1 #8790)
 id <01HNPFR0P5OW9D4GAS@EXAMPLE.COM> for ALONSO@EXAMPLE.COM; Fri,
 23 Aug 2012 16:48:41 -0400 (PDT)
Date: Fri, 23 Aug 2012 16:48:40 -0400 (PDT)
From: Prospero <prospero@example.com>
To: Prospero@example.com
Subject: Sea Voyage
Message-id: <01HNPFR12JYA9D4GAS@example.COM>
MIME-version: 1.0
Content-type: TEXT/PLAIN; CHARSET=US-ASCII
Content-transfer-encoding: 7BIT

Message content:
----------------------------------------------------------------------
Interested in joining me for a short sea voyage?

              - Alonso
qm.user>
```

return

Return a message to its sender.

SYNTAX

return *[message-id[,...]]*

Qualifiers	Defaults
`-all`	`-noall`
`-confirm`	`-noconfirm`
`-log`	`-log`

PARAMETERS

message-id
A comma separated list of one or more message identification numbers shown with a previous `directory` command. Ranges are allowed.

DESCRIPTION

Queued messages may be returned to their originator with the `return` command. The messages to be returned are specified by their message identification numbers shown by the most recent `directory` command. That number appears in the leftmost column of the `directory` command listing.

QUALIFIERS

`-all`
`-noall` ***(default)***
Return all messages shown by the last `directory` command. Unless `-noconfirm` is specified with `-all`, you will be required to confirm any `return -all` operation.

`-confirm`
`-noconfirm` ***(default)***
When `-confirm` is specified, you will be prompted to confirm each message return operation.

`-log` ***(default)***
`-nolog`
Specifies whether informational messages for each message return operation are generated.

run

Execute commands from a file.

SYNTAX

run *file-spec*

Qualifiers	Defaults
`-ignore`	`-noignore`
`-log`	`-log`

restrictions Must be able to access the file and execute the commands.

PARAMETERS

file-spec
Required parameter specifying the file to execute.

DESCRIPTION

The `run` command causes `pmdf qm` to open the specified file and read and execute each line from it as a `pmdf qm` command. Unless `-ignore` is specified, command execution will be aborted should one of the commands generate an error. By default each command is echoed to the terminal before being executed; specify `-nolog` to suppress this echo.

QUALIFIERS

`-ignore`
`-noignore` ***(default)***
By default, command execution will be aborted should one of the commands generate an error. Specify `-ignore` if you want execution to continue even if an error occurs.

`-log` ***(default)***
`-nolog`
Specifies whether commands are echoed to the display before they are executed.

6.4 db: Manage Personal Aliases and Mailing Lists

pmdf db is a utility with which to create and manipulate an alias database. Alias databases may either be a personal alias database or a system alias database. As the format of PMDF alias database is used for other PMDF databases (*e.g.*, PMDF_DOMAIN_DATABASE), pmdf db can also be used to manipulate non-alias databases.

pmdf db is invoked by the command

```
% pmdf db
```

and may be exited by either typing Control-D or issuing the quit-program command.

The command recall and editing capabilities are provided by the open source software libedit (also known as editline). By default, the standard "vi" key bindings are defined. You can change various elements of the editing environment, such as using "Emacs" key bindings instead of "vi", by creating in your home directory a file called .editrc. See the editrc manpage for more information.

6.4.1 Aliases

Aliases have multiple uses with e-mail. Individual users typically use aliases as abbreviations. For instance, rather than remember John Doe's full e-mail address, an alias jd can be created so that mail sent to the address jd is properly sent using John Doe's full address (*e.g.*, jd573@hostc.example.com). System managers often use aliases in order to create valid mail addresses for non-existent users. For instance, an alias named postmaster might be created and equated with the username root so that incoming network mail for the user postmaster is routed to the root account. These are just two examples of the many practical uses of aliases.

The process of interpreting an alias is called “alias expansion”. In the two examples above, the aliases jd and postmaster expand, respectively, to jd573@hostc.example.com and root.

In this documentation, the expansion of an alias is represented with the following notation

alias-name → alias-value

For example, John → jd573@hostc.example.com.

PMDF alias names are “case insensitive”. This means that the alias names jd, JD, jD, and Jd are all considered to be identical by PMDF; the case (upper *versus* lower case) of the individual characters in an alias name is irrelevant to PMDF. However, PMDF does preserve the case of alias values.

PMDF aliases may expand to:

— an address: jd → jd573@hostc.example.com,

— a list of addresses: staff → bob@example.com,sue@example.com,

— other aliases: jd → johnd → jd573@hostc.example.com,

— a list of aliases: all → staff,admin,faculty, or

— a mixture of addresses and aliases: list → staff,bob@example.com.

Note that in the above example, it is not clear whether or not the expanded value of an alias is another alias or not; *i.e.*, in “jd → johnd”, johnd could have been either an alias or a legitimate username. PMDF always starts by assuming that an address without any domain part (*e.g.*, @example.com) is an alias and attempts to expand it. When expanding an alias, PMDF first tries to look up the alias in the user’s personal alias database and, if the alias is not found there, then PMDF consults system-level alias sources. After expanding an alias once, PMDF then tries to expand the result (or results in the case of a list). This expansion process is repeated until no more expansions are possible at which point the results are all assumed to be real mail addresses and not aliases.

By default, alias names may be from 1 to 80 characters long and their expansion values 0 to 252 characters. This corresponds to a “long” alias database file which is the type of file `pmdf db` normally creates.

6.4.2 Mailing Lists

With `pmdf db` you may create and maintain your own mailing lists. A mailing list is merely a collection of e-mail addresses with which you associate an alias. Or, looked at a little differently, a mailing list is an alias which expands to a list of e-mail addresses. When you address a mail message to the alias, it actually goes to all of the addressees listed in the mailing list. The act of sending a mail message to a mailing list is referred to as “posting”.

A mailing list is created in two steps:

1. Create a text file containing the list of e-mail addresses associated with the mailing list. Each address should be on a separate line in the file. The file itself is referred to as a `mailing list file`; the addresses in the file are the mailing list’s membership.
2. Choose an alias name, *`alias-name`*, to associate with the mailing list. Then, in `pmdf db`, issue the command

   ```
   db> add alias-name "<filename"
   ```

   `filename` should include a full file path specification.

After these steps have been taken, the mailing list is set up and ready to use.

For example, suppose the user sue@example.com wants to set up a mailing list named foo-list. The members of the mailing list will be bob@example.com, judy@example.com, ralph@stateu.edu, and sue@example.com herself. Sue first creates the mailing list file /usr/users/sue/sample.dis which contains the four lines

```
bob@example.com
judy@example.com
ralph@newcompany.com
sue@example.com
```

Then Sue establishes the alias foo-list for her own use as follows:

```
% pmdf db
db> add foo-list "</usr/users/sue/sample.dis"
db> show foo-list attributes
Key          Value
-----------  -----------------------------
sample-list     </usr/users/sue/sample.dis
Attributes:  private,no_expand,block_receipts,mail_address
[1 entries shown]
db>
```

At any time you may add or remove members from the mailing list. You do so by simply editing the mailing list file removing or adding addresses from or to it.

Note that only the owner of this list, user sue, may post to the list, and she may only post to it when submitting from a user agent on the PMDF system itself.

As another example, mailing lists defined in LDAP can also be used, for example:

```
db> add ldap_all_users <"""ldap:///dc=example,dc=edu?mail?sub?(cn=*)"""
```

Note the three double-quotes around the LDAP URL. This is required.

6.4.3 Advanced Mailing Lists

`pmdf db` allows you to control a variety of aspects of messages posted to a mailing list alias, including associating error return, reply to, and other special addresses with mailing lists. To use these features, an extended alias specification must be used when declaring the alias for the mailing list:

```
db> add alias-name "<filename, named-parameters, error-return-address,
    reply-to-address, errors-to-address, warnings-to-address, comments"
```

The *named-parameters* item is described in Section 6.4.3.1; the remaining items, called “positional parameters”, are described in Section 6.4.3.2.

The two positional parameters *error-return-address* and *reply-to-address* are two particularly useful items. You are strongly encouraged to use the *error-return-address* parameter so as to control where error messages concerning postings to your list are directed. You can use the *reply-to-address* parameter to make the preferred reply address some specified address.

6.4.3.1 Named Parameters

Named-parameters are used to associate options with a mailing list. There can be zero or more named parameters, each separated by commas, and they must appear before any positional parameters. The general syntax of a named-parameter is:

```
[name] value
```

Here *name* is the name of the parameter and *value* is its corresponding value. The square brackets are a mandatory part of the syntax: they do not indicate an optional field.

The available named parameters are:

BLOCKLIMIT
LINELIMIT

The BLOCKLIMIT and LINELIMIT parameters may be used to limit the size of messages that may be posted to the list. The *value* item must be an integer number of PMDF blocks, for [BLOCKLIMIT], or an integer number of lines, for [LINELIMIT]. The size of a PMDF block is normally 1024 bytes. The default value for these parameters is 0, meaning that no limit is imposed on the size of message that may be posted to the list (apart, that is, from any system wide limits).

DELAY_NOTIFICATIONS
NODELAY_NOTIFICATIONS

The DELAY_NOTIFICATIONS named parameter requests that NOTARY delay notifications be sent for mailing list postings; the NODELAY_NOTIFICATIONS named parameter requests that NOTARY delay notifications not be sent for mailing list postings. The *value* specification is currently ignored and should always be NONE.

HEADER_ADDITION

HEADER_ADDITION may be used to specify a file of headers to be added to posted messages. The argument must be a full file specification for the file containing headers to be added.

In particular this facility can be used to add the standard mailing list headers defined in RFC 2369. For instance, a user amy@example.com that has set up a list named listname might use a header addition file along the lines of the following:

```
List-Help: <mailto:amy@example.com?subject=help%20on%20listname>
List-Subscribe: <mailto:amy@example.com?subject=subscribe%20listname>
List-Unsubscribe: <mailto:amy@example.com?subject=unsubscribe%20listname>
List-Post: <mailto:amy@example.com>
List-Owner: <mailto:amy@example.com?Subject=listname>
List-Archive: <mailto:amy@example.com?subject=request%20listname%20archive>
```

IMPORTANCE
PRECEDENCE
PRIORITY
SENSITIVITY

The IMPORTANCE, PRECEDENCE, PRIORITY, and SENSITIVITY named parameters are used to generate respective headers on messages posted to the list; the *value* specification is inserted on the respective header line.

SEQUENCE_PREFIX
SEQUENCE_SUFFIX
SEQUENCE_STRIP

The SEQUENCE_PREFIX and SEQUENCE_SUFFIX named parameters request that a sequence number be prepended or appended to the Subject: lines of messages posted to the list. The *value* item gives the full file path specification of a sequence number file. This file is read, incremented, and updated each time a message is posted to the list. The number read from the file is prepended, in the case of SEQUENCE_PREFIX, or appended, in the case of SEQUENCE_SUFFIX, to the message's Subject: header line. This mechanism provides a way of uniquely sequencing each message posted to a list so that recipients can more easily track postings and determine whether or not they have missed any.

By default, a response to a previously posted message (with a previous sequence number) retains the previous sequence number as well as adding a new sequence number to the subject line; the build up of sequence numbers shows the entire "thread" of the message in question. However, the SEQUENCE_STRIP named parameter can be used to request that only the highest numbered, *i.e.*, most recent, sequence number be retained on the subject line. The *value* item is currently ignored and should always be NONE.

Important note: To ensure that sequence numbers are only incremented for successful postings, a SEQUENCE_PREFIX or SEQUENCE_SUFFIX named parameter should always appear as the last named parameter; that is, if other named parameters are also being used, the SEQUENCE_ named parameter should appear at the end of the list of named parameters.

Sequence number files are binary files and must have the proper file attributes and access permissions in order to function correctly. In particular, sequence number files must be writeable from the perspective of the PMDF user account, normally pmdfuser.

To create the file *seq-file-spec* for use as a sequence number file, issue the command:

```
% touch seq-file-spec
```

or

```
% cat >seq-file-spec
```

You will then need to have your system manager allow the pmdfuser account access to the file by setting the userid and groupid for the file to the values for the pmdfuser account.

TAG

The TAG named parameter may be used to prefix specified text to the Subject: header of posted messages. The value item should be the string to be added.

USERNAME

The USERNAME named parameter may be used to set the username that PMDF will consider to "own" these mailing list messages. For instance, the pmdf qm utility will allow that username to inspect and bounce messages in the queue resulting from expansion of this mailing list. The value item should be the username of the account to "own" the mailing list postings.

6.4.3.2 Positional Parameters

With one exception, the positional parameters in a mailing list specification provide alternate addresses to which certain sorts of list related activity should be directed (*e.g.*, an address to which errors should be sent to rather than back to the list itself).

The positional parameters are so named for a reason: their position in the comma separated list distinguishes which parameter is being specified. When more than one parameter (positional or otherwise) is specified, they must be separated by commas. If you want to specify a positional parameter but omit some which come first, then specify asterisks, `*`, for the positional parameters which you want to omit. For example,

```
db> add sample-list "</usr/users/sue/sample.dis, *, *, sue@example.com"
```

Finally, to make the use of a positional parameter conditional, end the parameter value with an asterisk. In this case the value associated with the parameter will only be used if the corresponding message header line is not present in the message being posted to the list. (The asterisk will not appear in the message header should the parameter take effect.)

Without further ado, the positional parameters are:

error-return-address

error-return-address specifies an address to replace the message's regular envelope `From:` address as well as an address to be inserted into the header as an `Errors-to:` address. This header line is not generated if this address is not specified.

reply-to-address

The *reply-to-address* parameter specifies an address to be used as a Reply-to: address.

errors-to-address

The *errors-to-address* parameter specifies an address to be placed on the `Errors-to:` header, if this address should be different from the *error-return-address* that is used as the envelope `From:` address.

warnings-to-address

The *warnings-to-address* parameter specifies an address to be placed on the `Warnings-to:` header line. This header line is not generated if this address is not specified.

comments

The *comments* parameter specifies a string to be placed in a `Comments:` header line. This header line will add to any `Comments:` header lines already present in the message being posted to the list.

6.4.3.3 Examples

In this example, the user sue@example.com sets up a mailing list named foo-list. The mailing list file is the file `/usr/users/sue/sample.dis` and its contents are shown in Example 6–1. The commands used to set up the list are shown in Example 6–2. In that example, the add command must be entered as one line; it is shown broken into two lines for typographical reasons only. sue@example.com may post to her list by sending to the address sample-list@example.com from a user agent on the PMDF system (a user agent that invokes the PMDF sendmail replacement).

Two positional parameters, *error-return-address* and *comments*, are specified. The `error-return-address` parameter specifies that error messages associated with the list should be sent to sue@example.com; the `comments` parameter generates a

Comments: header line reading “Sue's sample list”. which will appear in each posting to the list.

Example 6–1 Sample Mailing List: The Mailing List File

```
bob@example.com
judy@example.com
ralph@example.com
sue@example.com
```

Example 6–2 Sample Mailing List: Declaring the Alias

```
% pmdf db
db> add foo-list "</usr/users/sue/sample.dis,
             sue@example.com,*,*,*,Sue’s sample list"

db> exit
```

6.4.3.4 Length Restriction on List Definitions

Keep in mind the length limit of alias expansion values of 252 characters when defining a more sophisticated mailing list with multiple parameters. Most lists can be suitably defined with just a few of the possible mailing list parameters discussed above. But if you have a list for which you really want to use a lot of parameters, then you may need to define the list in stages.

For instance, to define a list friends-list that has HEADER_ADDITION, NODELAY_NOTIFICATIONS, SEQUENCE_PREFIX, USERNAME, and IMPORTANCE named parameters, as well as error-return-address and comments positional parameters, the list can be defined in two stages, using a subsidiary friends-list-stage2 definition, *e.g.*,

```
db> add friends-list "</usr/users/alan/friends-list.dis,
[HEADER_ADDITION] /usr/users/alan/friends-list-headers.txt,
[NODELAY_NOTIFICATIONS] NONE, [SEQUENCE_PREFIX] /usr/users/alan/friends.seq,
[USERNAME] alan"
db> add friends-list-stage2 "</usr/users/alan/friends-list-stage2.dis,
[IMPORTANCE] High, alan@example.com, *, *, *, A chatty message list for Alan’s
friends -- contact Alan at 555-1212 for more information"
```

where the /usr/users/alan/friends-list-stage2.dis file contains just the line:

```
friends-list
```

and the /usr/users/alan/friends-list.dis contains all the actual recipient addresses.

6.4.4 Operation of db

`pmdf db` prompts for input with a `db>` prompt. Typing a control-D at any point while entering a command will cause `pmdf db` to immediately stop execution. The quit-program command will also cause `pmdf db` to stop execution.

The rest of the command line after the `pmdf db` will be scanned for a `pmdf db` command. Placing a command on the invocation line is optional; if one is specified `pmdf db` will terminate after the last one has been executed. If no command appears `pmdf db` will operate by prompting the user for commands.

When first invoked, `pmdf db` will open your personal alias database, `~/aliasesdb.*`.

While entering `pmdf db` commands to the `db>` prompt, the following command interaction features are available:

- Command abbreviation: commands may be abbreviated to their simplest, unambiguous form.
- Input files: command files may be input and executed by using the command `run` *`infile`* with *`infile`* the name of the file to input.

6.4.5 Commands

The following sections provide full descriptions of all `pmdf db` commands.

6.4.5.1 add

Syntax: add *alias-name alias-value [attributes[,...]]*

With the add command an alias and its expansion value may be added to the database currently opened. If the alias expansion value contains any spaces, commas, or upper case characters which should not be converted to lower case, then the expansion value must be enclosed in double quotes. Any double quotes in the expansion value must be “doubled” (*i.e.,* entered as two consecutive double quotes). Examples of these two cases are:

```
db> add staff "bob@example.com,sue@example.com,tom@example.com"
db> add jo """John Owen""@example.com"
db> add ldap-all <"""ldap:///dc=example,dc=edu?mail?sub?(cn=*)"""
```

The first add command establishes the alias staff → bob@example.com,sue@example.com,tom@example.com

This alias required quoting since it included commas. The second add command establishes the alias

jo → "John Owen"@example.com

and was enclosed in double quotes because of the space in it, and the need to not convert John Owen's name to "john owen". Also, the double quotes present in the alias expansion value were doubled; that is, each double quote, ", was specified as two double quotes, "".

The third add command establishes the alias

ldap-all → <"ldap:///dc=example,dc=edu?mail?sub?(cn=*)"

Which means that PMDF will expand the alias into a mailing list by performing an LDAP query using the LDAP URL specified.

As another example, consider entering the alias JD with the simple expansion value jd573@hostc.example.com:

```
db> add JD jd573@hostc.example.com
[Entry added to database]
db> show JD
Key         Value
----------  ----------------------------
jd          jd573@vaxc.example.com
[1 entry shown]
db>
```

Note that the alias name along with its translation value was converted to lower case. `pmdf db` will always translate the name of an alias to lower case; PMDF does not do case sensitive alias matching. To prevent the alias translation value from being converted to lower case, enclose it in double quotes; *e.g.*,

```
db> modify JD "JD573@HOSTC.EXAMPLE.COM"
[1 entry modified]
db> show JD
Key         Value
----------  ----------------------------
jd          JD573@HOSTC.EXAMPLE.COM
[1 entry shown]
db>
```

The optional *attributes* parameter of the add command may be one or more comma separated keywords selected from the list:

mail_address

Treat this alias as a mail address; *i.e.*, set the mail address attribute flag for this alias. Default when in normal mode. Not set when an `override on` command has been issued.

non_mail_address
Do not treat this alias as a mail address; that is, clear the mail address attribute flag for this alias.

expand
This attribute forces the expansion of the alias into its component addresses. All of the component addresses will be listed in the message header as the alias expands.

no_expand
This attribute inhibits the expansion of the alias into its component addresses. The alias itself will appear in the header in some form; see the description of the public and private attributes above for details on the forms the alias can take.

pass_receipts
This attribute enables the passage of requests for delivery and read receipts through to all recipients as the alias is expanded. Note that allowing the passage of such a request to a large distribution list may result in a lot of return mail.

block_receipts
This attribute disables the passage of requests for delivery and read receipts through to all recipients as the alias is expanded. Requests for such receipts are honored at the time the alias expands; in effect the alias is treated as the message destination.

When an alias is entered with the add command at the `db>` prompt, the alias will automatically be given the mail address attribute unless an `override on` command has been previously issued or the `non_mail_address` keyword is specified with the `add` command.

6.4.5.2 close

Syntax: close

The `close` command closes the currently open alias database. Use the `open` command to open a different database. `pmdf db` will automatically close any open database when the `exit` or `quit` commands are issued.

6.4.5.3 copy

Syntax: copy *from-alias-name to-alias-name*

The `copy` command creates a new alias with the name *to-alias-name* and associates to it the expansion value of the alias with the name *from-alias-name*. A subsequent change to the "from" alias will not affect the "to" alias. Any attributes associated with the "from" alias will be copied to the "to" alias regardless of whether or not an `override on` command has been issued previously.

```
db> add postmaster "root@thor.example.com"
[Entry added to database]
db> copy postmaster postmast
[1 entry copied]
db> show post*
Key         Value
----------  -----------------------------
postmast    system@thor.example.com
postmaster  system@thor.example.com
[2 entries shown]
db>
```

6.4.5.4 exit

Syntax: exit-program

The exit and quit commands are identical and each causes `pmdf db` to close any open database and then exit.

6.4.5.5 help

Syntax: help *[topic]*

Obtain help on a topic.

6.4.5.6 modify

Syntax: modify *alias-name new-alias-value*

The `modify` command is used to replace the expansion value of an alias with a new expansion value.

```
db> add postmaster "root@thor.example.com"
[Entry added to database]
db> show postmaster
Key         Value
----------  -----------------------------
postmaster  root@thor.example.com
[1 entry shown]
db> modify postmaster "ariel@example.com"
db> show postmaster
Key         Value
----------  -----------------------------
postmaster  ariel@example.com
[1 entry shown]
db>
```

Wild cards may be used when specifying the alias name in order to modify one or more aliases simultaneously.

6.4.5.7 open

Syntax: open *database-name* [huge | long | short]

The open command opens an alias database after first closing any currently opened database. If the database to be opened already exists, then pmdf db will automatically determine whether or not the database is a "huge" (stores 80/1024 character long alias names/values), or "long" (stores 80/132 character long alias names/values) or "short" (stores 32/80 character long alias names/values). If the database does not already exist, then it will be created.

When pmdf db is first invoked, your own personal alias database is automatically opened, or created if it doesn't exist.

6.4.5.8 override

Syntax: override on | off

The override command is useful when looking at or modifying databases other than an alias database. Ordinarily the database manipulation commands, add, copy, modify, rename, remove, set, and show will only operate on aliases with the mail_address attribute. When the command override on has been issued, these commands may be used to manipulate any entry in the database regardless of its attributes (or lack thereof).

The override off command negates the override on command.

6.4.5.9 quit

Syntax: quit

The exit and quit commands are identical and each causes pmdf db to close any open database and then exit.

6.4.5.10 remove

Syntax: remove *alias-name*

With the remove command, one or more aliases may be removed from the database. Wild cards may be used when specifying aliases to be removed. For instance, to remove all aliases from a database, issue the command remove*.

6.4.5.11 rename

Syntax: rename *old-alias-name new-alias-name*

The rename command is used to rename an alias without altering its expansion value:

```
db> add postmaster "root@thor.example.com"
[Entry added to database]
db> show postmaster
Key         Value
----------  -----------------------------
postmaster  root@thor.example.com
[1 entry shown]
db> rename postmaster post
[1 entry renamed]
db> show post
Key         Value
----------  -----------------------------
post        root@thor.example.com
[1 entry shown]
db>
```

6.4.5.12 run

Syntax: run *file-spec*

The run command may be used to execute a file of pmdf db commands.

6.4.5.13 set

Syntax: set *alias-name attributes[,...]*

The set command may be used to grant or remove attributes from aliases. The alias name specification may include wild cards. The allowable attribute names are listed in the add command description described in Section 6.4.5.1.

6.4.5.14 show

Syntax: show [*alias-name* [attributes]]

The show command is used to list the contents of a database. The optional attributes keyword, when supplied, causes the attributes associated with each alias to also be displayed:

```
db> add postmaster "root@thor.example.com"
[Entry added to database]
db> show postmaster attributes
Key         Value
----------  -----------------------------
postmaster  root@thor.example.com
Attributes: private,expand,block_receipts,mail_address
[1 entries shown]
db>
```

The alias name specification may contain wild cards. To see all entries with the mail_address attribute, issue the command show; to see absolutely all entries, first issue the command override on followed by the command show

6.4.5.15 wildcards

Syntax: wildcards ignore | interpret

By default, the characters * and % in alias names are interpreted as wildcards: an asterisk, *, will match zero or more characters while each percent sign, %, will match precisely one character. The command wildcards ignore will cause pmdf db to not interpret asterisks or percent signs as wildcards; the command wildcards interpret will resume interpretation of wildcards.

6.4.5.16 write

Syntax: write *file-name* [alias | pine]

The write command is used to create a command file which, when fed back into pmdf db with the run command, will recreate the entire database. If the alias keyword is specified, then a PMDF alias file (using PMDF's alias file format) will instead be produced. If the pine keyword is specified, then a pine addressbook is written.

For instance, the following commands will create a database named db2 which duplicates the database db1:

```
db> open db1
db> write makedb1
db> open db2
[creating database]
db> run makedb1
db>
```

The above example presupposes the existence of a database named db1. The file makedb1 created with the write command is an ordinary text file which may be edited with any text editor.

Note that this is not an efficient way to duplicate a database — a shell command such as the `cp` command will work much more quickly. The `write` command is intended as a means of creating a textual representation of a database which can be edited as a text file and later turned back into a database.

7 Notes for POP and IMAP Clients

This chapter provides a few tips for POP and IMAP client users; that is, users who use POP or IMAP clients remotely to access a mailbox on the PMDF server system. The remote system may be a PC, Macintosh, UNIX, OpenVMS, or other sort of computer system—as long as the remote system has a POP or IMAP client on it and TCP/IP access to a PMDF system running a POP or IMAP server, respectively, the user can access their messages on the PMDF server.

7.1 The pmdf password Utility: Enabling Authentication (login) Mechanisms

The `pmdf password` utility is used to add and change password values that may be used for special authorization purposes, such as authentication by POP and IMAP clients.

Whether you need to use this utility will depend on your site's configuration, (and on what POP or IMAP client you use and what underlying commands it uses). If you access the native BSD message store from POP or IMAP clients, then you may need to. Otherwise (for instance, if you access the PMDF popstore from a POP client or the PMDF MessageStore from an IMAP or POP client) then you probably will not.

APOP passwords, used by some POP clients such as Eudora, and CRAM-MD5 passwords, used by some POP and IMAP clients, cannot be stored in the system password file. Therefore, in order to support use of the POP protocol's APOP command or AUTH command with CRAM-MD5, or the IMAP protocol's authenticate command with CRAM-MD5, you must have a password entry stored in another authentication source such as the PMDF password database.

When using the PMDF password database as the source of authentication information, note that it may contain several entries, one for each allowed service value. The sort of connection (for instance, whether POP or IMAP) will control which service entry is preferentially checked. Queries by the POP server will first check your POP service entry, but if such an entry does not exist will fall through to your `DEFAULT` service entry. Queries by the IMAP server will first check your IMAP service entry, but if such an entry does not exist will fall through to your `DEFAULT` service entry.

The use of service specific password database entries is not typical; typically, you would simply have one entry, your `DEFAULT` service entry, used whenever the PMDF password database is queried. But if you do want to use service specific password database entries, while the above description of service specific probes may sound complicated, the goal is simply to query the "natural" password entry for each case.

To set your password (for the DEFAULT service) in the PMDF password database, issue the command:

```
% pmdf password
```

You will then be prompted to enter your password. For further details, see the discussion of pmdf password in Chapter 6.

7.2 The "Do not delete this message" Message in the Berkeley Mailbox

The native Berkeley mailbox does not have certain features that IMAP or POP access require. So if you access your Berkeley mailbox from an IMAP or POP client, the PMDF IMAP or POP server will create a pseudo-message at the start of your folder to contain the additional details that the Berkeley mailbox itself doesn't have. This pseudo-message by default says "Do not delete this message", (though your system administrators may have modified the text for your site). You should not delete this message; deleting it while you have another IMAP or POP session open to your mailbox may mean that the other session cannot function properly; in any case, the message will be recreated the next time you use IMAP or POP to access your mailbox.

7.3 IMAP Access to Hierarchical Folders

IMAP and IMAP clients permit so-called *hierarchical folders*. The forward slash character, /, is used as the hierarchy separator, just as when specifying subdirectories and files. And indeed, for the native Berkeley mailbox, hierarchical folders are implemented as UNIX directories and files. That is, for the native Berkeley mailbox an IMAP hierarchical folder /apple/banana/carrot will correspond to the file carrot within a subdirectory /apple/banana/ under your default directory. Note that hierarchical folders in PMDF MessageStore mailboxes do not correspond to UNIX subdirectories in this way.

It is not necessary to create "intermediate" folders before creating a terminal folder; for instance, folder /apple/banana/carrot may be created even if the folder (which in the case of a native Berkeley mailbox is implemented as a subdirectory) /apple/banana/ did not previously exist; the IMAP server will create the necessary folder (the necessary subdirectory in the case of a native Berkeley mailbox) for you automatically.

Note that since "intermediate" folders for the native Berkeley mailbox are actually UNIX subdirectories, it is not possible to store messages in such "intermediate" folders; messages may only be stored in a file at the "end" of a folder specification. But with PMDF MessageStore mailbox hierarchical folders, as they are not limited by subdirectory issues, it *is* possible to store messages in "intermediate" folders.

7.4 Subaddresses/Folders in MessageStore Addresses

You may include extra information in your mail address. This extra information is referred to as a “subaddress”. Any material following a plus, +, in your address is considered to be a subaddress. For instance, “test” in the address bob+test@example.com is a subaddress.

A subaddress in a PMDF MessageStore address is considered to describe the folder to which the message should be delivered. For instance, for the PMDF MessageStore user bob@example.com, a message addressed to bob+test@example.com would be delivered to bob’s folder named “test”.

7.5 Eudora’s Password Changing Command

PMDF supports the Eudora POP client’s adhoc method for password changing. Check with your system administrators if you have trouble using the Eudora password changing command; it may be that your system administrators have not started the PMDF POPPASSD server to enable this feature.

Index