Message Exchange User's Guide

August, 2008

This manual provides information for users of Message Exchange, electronic mail software for VMS systems.

Revision/Update Information:	This is a revised manual.
Operating System and Version:	OpenVMS Alpha [tbd] or later
	OpenVMS Industry Standard 64 [tbd] or later
Software Version:	Message Exchange V6.0

Matthew Madison
Santa Cruz, California

27 August 2008

Copyright ©2008 Matthew Madison.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright owner nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of the software were adapted from other open source projects. Refer to the Release Notes for copyright and license information.

Contents

PREFACE v

CHAPTER 1 USING MESSAGE EXCHANGE WITH VMS MAIL 1–1

1.1 SPECIFYING AN ADDRESS 1–1
1.1.1 Displaying MX Address Translations 1–1
1.1.2 Multiple Recipients 1–2
1.1.3 Quotation Marks 1–2

1.2 USING SET FORWARD WITH MX 1–2

1.3 PERSONAL NAME 1–3

1.4 SIGNATURE FILES 1–3
1.4.1 Automatic Signature Inclusion 1–3

1.5 REDIRECTING REPLIES 1–4

1.6 DIRECTING DELIVERY TO VMS MAIL FOLDERS 1–4
1.6.1 Folder Delivery Address Format 1–4
1.6.2 Format of MX_FOLDERS.DAT 1–5
1.6.3 Example MX_FOLDERS.DAT 1–5
1.6.4 Specifying Folders in From Addresses 1–5

1.7 DELIVERY STATUS NOTIFICATIONS 1–6
1.7.1 Notification Settings 1–6
1.7.2 Returned-Information Settings 1–7
1.7.3 Combining the Settings 1–7
1.7.4 Limitations 1–7

1.8 NETWORK DELIVERY DELAYS 1–7
1.8.1 Displaying MX Informational Messages 1–8

1.9 SENDING BINARY FILES TO OTHER VMS USERS 1–8

Contents

CHAPTER 2 THE MXALIAS UTILITY 2–1

2.1 ADDING AN MX ALIAS 2–1

2.2 USING AN MX ALIAS 2–2
2.2.1 Disabling System-Wide MX Alias Lookups 2–2
2.2.2 Displaying MX Address Translations 2–2

2.3 DISPLAYING ALIASES 2–3

2.4 MODIFYING ALIASES 2–3

2.5 REMOVING ALIASES 2–3

CHAPTER 3 ELECTRONIC MAILING LISTS 3–1

3.1 TYPICAL INTERNET MAILING LISTS 3–1

CHAPTER 4 MAIL-BASED FILE SERVERS 4–1

4.1 GET HELP 4–1

4.2 MX FILESERV COMMANDS 4–1
4.2.1 Packages 4–2
4.2.2 Binary Files 4–2

APPENDIX A MESSAGE HEADER FORMAT A–1

A.1 VMS MAIL HEADERS A–2
A.1.1 From Header A–2
A.1.2 To and CC Headers A–3
A.1.3 Subject Header A–3

Preface

Message Exchange (MX) is software that provides store-and-forward routing and delivery of electronic mail messages. It can also provide mailing list and file distribution services. MX can be used to enhance local electronic mail (E-mail) support, and it can be used with several kinds of network protocols to provide a unified E-mail interface to different networks.

Intended Audience

This manual is intended for any VMS MAIL user who uses MX, and users of MX's mailing list and file distribution services. The reader should already know the basics of using VMS and the VMS MAIL utility.

Document Structure

This guide consists of four chapters and one appendix.

Chapter 1	Describes the MX/VMS MAIL interface.
Chapter 2	Describes the MXALIAS utility.
Chapter 3	Describes the mailing list handler.
Chapter 4	Describes the file server.
Appendix A	Describes MX message formats in detail.

Related Documents

You can find additional information in the following documents:

- *Message Exchange Installation Guide* describes the installation of MX.
- *Message Exchange Management Guide* describes the management and operation of MX.
- *Message Exchange Mailing List/File Server Guide* describes the management and operation of the MX mailing list and file server.
- *Message Exchange Release Notes* contain information and updates not included in this manual. The release notes are part of the software distribution kit.
- *VMS Mail Utility Manual* describes the VMS MAIL utility in detail.

1 Using Message Exchange with VMS MAIL

Message Exchange (MX) interfaces with VMS MAIL to provide the means for addressing outgoing mail through MX. It also ensures that mail that is delivered via MX has an appropriate source address for replies, and provides support for signature files and user-specified reply-to addresses.

1.1 Specifying an Address

MX interfaces with VMS MAIL as a ''foreign protocol''. When using VMS MAIL, you address mail to be sent through MX by specifying an address of the form:

```
MX%"user@host"
```

The leading MX% tells VMS MAIL to invoke the MX protocol handler; the address, which should be surrounded by quotation marks to prevent the address from being converted to upper case and prevent the @-sign from being interpreted by VMS MAIL, is the network mail address of the user you wish to send mail to.

If the user is on the local host, you can omit the @*host* part of the address, and the quotation marks, just specifying

```
MX%username
```

for an address.

The MXALIAS utility can be used to define MX aliases for e-mail addresses; see Chapter 2, The MXALIAS Utility, for information about using MXALIAS. MX aliases are used just as if sending mail through MX to a local user:

```
MX%alias
```

Any MX% address given without the @*host* part of the address is checked to see if it is an MX alias. If it is, the equated address is used; if not, the specified address is assumed to be that of a local user.

1.1.1 Displaying MX Address Translations

If you want to see all address translations made by MX for MX% addresses passed from VMS Mail, you can define the logical MX_VMSMAIL_SHOW_ ADDR as shown in the following command:

```
$ DEFINE MX_VMSMAIL_SHOW_ADDR TRUE
```

If the logical is defined, MX displays the final address used for a given address:

```
MAIL> SEND
To:     MX%JOE, MX%"MX-List@WKUVX1.WKU.EDU", SYSTEM
  MX rewrote alias JOE as <SYSTEM@WKUVX1.WKU.EDU>
  MX rewrote MX-List@WKUVX1.WKU.EDU as <MX-List@WKUVX1.WKU.EDU>
Subj:   ....
```

Note that ''SYSTEM'' was not passed to MX because it was not specified with the MX% prefix. Also note that JOE had been defined as an alias equal to SYSTEM@WKUVX1.WKU.EDU using the MXALIAS utility (described in Chapter 2).

Placing the MX_VMSMAIL_SHOW_ADDR logical definition in your LOGIN.COM will cause MX to always show you all address translations.

1.1.2 Multiple Recipients

When sending messages to more than one recipient through MX, each recipient's address requires the MX% prefix (and quotation marks, if needed). For examples:

```
MAIL> SEND
To: SMITH, MX%"jones@otherhost.edu",BROWN,MX%NAMES-L
```

Note that you can mix plain, local usernames with MX-directed addresses in the same message.

1.1.3 Quotation Marks

VMS MAIL cannot handle quotation marks within an address. MX works around this problem by substituting apostrophes instead. For example, if the destination address is

```
"node::user"@remote.host
```

you can specify this address in VMS MAIL as

```
MX%"'node::user'@remote.host"
```

To enter an apostrophe in an address, quote the apostrophe with a backslash. For example, if the destination address is

```
o'reilly@remote.host
```

you would enter

```
MX%"o\'reilly@remote.host"
```

1.2 Using SET FORWARD with MX

You can use the SET FORWARD command in VMS MAIL to set a forwarding address for your mail through MX. To do this, however, requires that you add extra quotes to the address. The forwarding address should be quoted, and, since MX addresses must be quoted, the inner quotes must be doubled to comply with the command parsing. For example:

```
MAIL> SET FORWARD "MX%""user@host"""
```

You should be sure to check the forwarding address with SHOW FORWARD and to send yourself a test message to ensure that you specified the address correctly.

1.3 Personal Name

The SET PERSONAL_NAME command in VMS MAIL lets you enter your real name, to be appended to your VMS username on outgoing mail. Messages sent via MX will also include your personal name if you have one set.

1.4 Signature Files

The MX/VMS MAIL interface provides support for ‘‘signature’’ files. A signature file is a file that contains your name, E-mail address, and any other information that you would like to have included in your outgoing mail messages. It should be no more than a few lines long and should probably contain lines that do not exceed 80 characters in length. For example:

```
Peter Shandy, Ph.D.
Horticulture Department
Balaclava Agricultural College
shandy@buster.balaclava.edu
```

Once you create a signature file, you inform MX of its existence by defining the logical name MX_SIGNATURE:

```
$ DEFINE MX_SIGNATURE device:[directory]name.type
```

You can then have the signature included in your message by entering the line

```
/SIGNATURE
```

in your message. To be recognized, there can be no other text on the line and no leading blanks. Case is not important, and you can abbreviate SIGNATURE to SIG. Your signature file will be inserted in the message at the point where you place the /SIGNATURE line.

Note that the signature is included only in copies of the message that are sent via MX; if you also send your message to users not using the MX% prefix, they will just see the /SIGNATURE line and not your signature file.

To enable your signature file every time you login, include the DEFINE command in your login command procedure.

1.4.1 Automatic Signature Inclusion

Your signature file can be included automatically at the end of your message by defining the logical name MX_AUTO_SIGNATURE:

```
$ DEFINE MX_AUTO_SIGNATURE text
```

The *text* is not important; as long as the logical name is defined, the signature file you specify with MX_SIGNATURE will will automatically be appended to the end of all subsequent MX messages. A /SIGNATURE line can be used to place the signature anywhere in the message (overriding the automatic appending).

If you wish to prevent the automatic inclusion of your signature file, enter a line

```
/NOSIGNATURE
```

in your message. The same formatting rules apply as for /SIGNATURE.

1.5 Redirecting Replies

Normally when you send a message via MX from your VMS account, the message will include information that will direct any replies to the message back to your VMS account. If you would rather have replies go to a different account, or to an account on a different system, you can define the logical name MX_REPLY_TO to include this information in the message:

```
$ DEFINE MX_REPLY_TO "user@host"
```

Note that you should not include the MX% prefix on the address, and you should not change quotation marks to apostrophes when you specify the address.

To have this reply address included in your messages every time you login, include the DEFINE command in your LOGIN.COM file.

Some mailers, including MX, allow multiple addresses on the ‘‘From:’’ line for messages. You can include multiple addresses in the MX_REPLY_TO definition to allow replies to be returned to multiple addresses (assuming the remote mailer allows it). For example, if you want replies to your messages to go to two different accounts, you could define the logical as follows:

```
$ DEFINE MX_REPLY_TO "user@host,user2@host2"
```

1.6 Directing Delivery to VMS MAIL Folders

MX normally delivers incoming messages to your NEWMAIL folder, just as VMS MAIL does for local messages. If you receive a large amount of e-mail, particularly from different mailing lists, MX’s *folder delivery* option may help you to keep your e-mail better organized.

1.6.1 Folder Delivery Address Format

MX accepts local e-mail addresses of the form *username+foldername*, and treats the characters appearing after the plus sign (+) as either the name of a folder or an alias for a folder name. You control the folder names into which MX can deliver messages by creating a file called MX_FOLDERS.DAT in your login directory.

When MX receives a message for you that contains a plus sign, it reads the contents of your MX_FOLDERS.DAT file. If it finds the folder name in the file, it delivers the message into that folder. Otherwise, it simply delivers the message to your NEWMAIL folder.

1.6.2 Format of MX_FOLDERS.DAT

The file MX_FOLDERS DAT *must* reside in your login directory (SYS$LOGIN:) and should be an ordinary VMS text file. Blank lines in the file are ignored. Other lines should contain one of the following:

- A folder name (may be specified with or without the leading plus sign).
- A folder alias specification of the form (the leading plus sign is *required*):

  ```
  +alias  actual-folder-name
  ```

- A comment, which is a line beginning with an exclamation point (*!*).

Leading and trailing blanks (and tabs) are ignored. Blanks and/or tabs separate alias names from actual folder names.

Since the folder specifications allowed in addresses are limited to only ASCII alphanumeric (A through Z, 0 through 9) characters, underscores (_), and hyphens (-), you can use folder aliases to allow delivery to folders that contain other characters that are allowed by VMS MAIL (such as dots and international 8-bit characters).

1.6.3 Example MX_FOLDERS.DAT

The following is an example of an MX_FOLDERS.DAT file for the user SAMPLE. It resides in SAMPLE's login directory.

```
!  MX-LIST for the MX-List mailing list
+mx-list
!  INFO-VAX for the Info-VAX mailing list
+info-vax
```

1.6.4 Specifying Folders in From Addresses

You can specify a folder name to be added to your e-mail address in outgoing messages by using the following directive on the *first line* of the text of the message:

```
/FROM=+foldername
```

This only applies to ordinary (non-/FOREIGN) text messages sent by VMS MAIL or DECwindows Mail. If present, MX's VMS MAIL interface will automatically remove the line from the text of the message and the address in the From: header to be *username+foldername*. Note that this folder name must contain only 7-bit ASCII alphanumeric characters, underscores, and hyphens.

If you use this directive when subscribing to a mailing list, and specify the folder name in your MX_FOLDERS.DAT file, then all messages from that mailing list will be delivered to the folder, rather to your default NEWMAIL folder.

1.7 Delivery Status Notifications

MX supports the Internet standard form of delivery status notifications (DSNs), which can provide information on the status of a message that you send. When a system supporting the DSN standards delivers a message, it examines the DSN control information you defined for the message and determines whether it should notify you about the status of the delivery.

To enable these notices, you must define the logical name MX_VMSMAIL_DSN_CONTROL. This logical name can be defined with up to two separate settings:

- a setting to specify when you want to be notified about the status of message deliveries, and
- a setting to specify how much of the message should be returned along with the delivery notice.

1.7.1 Notification Settings

The notification setting is specified as follows:

```
$ DEFINE MX_VMSMAIL_DSN_CONTROL "NOTIFY=type[,...]"
```

where *type* can be one of these notification types:

NEVER	Specifies that you do not want to be notified about the delivery of the message, even if it fails.
SUCCESS	Specifies that you want to be notified about successful delivery of the message, or if it has been successfully relayed to another system that does not support DSNs.
FAILURE	Specifies that you want to be notified if delivery of the message fails.
DELAY	Specifies that you want to be notified if delivery of the message is delayed for some reason.

If you specify NOTIFY=NEVER, you cannot specify any other notification type. Other types may be combined by separating the type keywords with commas.

Note that for SUCCESS notifications, notification of the successful delivery of the message may only be notification of a ‘‘relay’’ to another system. This happens if a remote system does not support DSNs, or the message is sent to a remote system using a protcol that is not capable of transmitting DSN requests.

1.7.2 Returned-Information Settings

The setting for returned information is either:

```
$ DEFINE MX_VMSMAIL_DSN_CONTROL "RETURN=HEADERS"
```

which specifies that you only want the RFC822 message headers included in the returned notification message, or:

```
$ DEFINE MX_VMSMAIL_DSN_CONTROL "RETURN=FULL"
```

which specifies that you want the entire message included in the returned notification message.

If you do not specify a RETURN setting, the system that issues the delivery status notification is allowed to choose whether it should return the entire message or just the headers.

1.7.3 Combining the Settings

You may specify both notification and returned-information settings by separating the two settings with one or more blanks. For example:

```
$ DEFINE MX_VMSMAIL_DSN_CONTROL "NOTIFY=SUCCESS,FAILURE RETURN=HEADERS"
```

This DSN control string specifies that you want to be notified if the message is delivered successfully or if it fails, and you want only the message headers included with the notification.

1.7.4 Limitations

Because DSN settings are controlled by a logical name, they apply to all recipients of all messages you send while the logical name is defined. You cannot specify per-message or per-recipient settings while using MX through VMS MAIL or DECwindows Mail.

1.8 Network Delivery Delays

Messages sent over any network can be delayed due to network outages, system loading, or other reasons. Once a message leaves the local system, there is no way to determine where the message may be held up. However, messages still on the local system awaiting network transfer can be displayed with the MAILQUEUE utility:

```
$ RUN MX_EXE:MAILQUEUE
```

MAILQUEUE lists any messages you have sent that are waiting for network transfer. All messages that cannot be sent are tried periodically, based on settings established by your system manager. If the number of attempts exceeds the established limit, the message is returned to sender with a message explaining why the transfer did not occur.

1.8.1 Displaying MX Informational Messages

If you want MX to display information messages indicating that your VMS Mail message has been successfully delivered to MX, you can define the logical MX_VMSMAIL_SHOW_INFO as shown in the following command:

```
$ DEFINE MX_VMSMAIL_SHOW_INFO TRUE
```

If the logical is defined, MX displays a line like the following when the message has been queued to MX:

```
%MX-I-MAIDLVR, message (entry number 22643) successfully delivered to MX
```

An informational message will also be displayed when a message is sent with SEND/FOREIGN:

```
%MX-I-BASE64, encoding MX foreign message using BASE64
```

Placing the MX_VMSMAIL_SHOW_INFO logical definition in your LOGIN.COM will cause MX to always display the informational messages.

1.9 Sending binary files to other VMS users

The VMS Mail command SEND accepts an undocumented qualifier, /FOREIGN. SEND/FOREIGN allows you to mail any VMS file to another user on the same system or over DECnet. The file retains all of the VMS file attributes. When the recipient tries to read the mail message containing the file, the following information is displayed:

```
    #2          14-APR-1993 15:28:02.11             NEWMAIL
From:   GOATHUNTER
To:     GOATHUNTER
CC:
Subj:   RESET.EXE

You cannot read this foreign format message
        Use the EXTRACT command to copy the message to an external file

MAIL>
```

The EXTRACT command copies the message to the named external file with all VMS file attributes intact.

The SEND/FOREIGN command can also be used to send VMS binary files through MX, if the target user is on a system running MX V3.3 or higher, MultiNet V3.3 or higher, or PMDF V4.1 or higher. When SEND/FOREIGN is used, MX encodes the message using an algorithm called BASE64, which is defined in RFC 1341, a document describing MIME (Multipurpose Internet Mail Extensions). The BASE64-encoded file is wrapped in a MIME-compliant message and mailed to the recipients. When the message is received on a system running the appropriate versions of either MX, MultiNet, or PMDF, the encoded binary file is automatically decoded and mailed to the local user as a foreign file. The recipient will receive two messages—one containing the headers for the message, and the other containing the foreign file as shown above.

The MIME ‘‘Content-Type:’’ for the file is ‘‘APPLICATION/VMS-RMS’’. MX will automatically recognize and decode incoming ‘‘VMS-RMS’’ files that are encoded using BASE64, as well as QUOTED-PRINTABLE files.

Note: The encoding done by MX is only compatible with the VMS mailers specified above. SEND/FOREIGN cannot be used to send binary files to non-VMS MIME-compliant mailers.

The following example demonstrates sending a binary file through MX:

```
$ mail

MAIL> send/noedit/foreign program.exe
To:     MX%"gene@KISS.COM"
Subj:   Here is that program I promised to send

Encoding MX foreign message using BASE64
Message (entry number 22244) successfully delivered to MX

MAIL>
```

Note: Non-VMS recipients or VMS recipients on systems *not* running the appropriate software will receive a single message containing the BASE64-encoded file. This message will most likely be meaningless for those recipients.

From the DCL prompt, the command MAIL/FOREIGN can be used to send a binary file to one or more recipients:

```
$ mail/foreign/subj="My LOGIN.COM" login.com "mx%""user@node.edu"""
```

2 The MXALIAS Utility

MXALIAS is a simple database manager for user-defined MX aliases. An alias is a name that is equated with a mail address that can be used to address electronic mail. For example, the address ‘‘BOB’’ can be equated with ‘‘smithjb@node1.school.edu’’; it can then be used in VMS Mail by specifying MX%BOB at the ‘‘To:’’ prompt:

```
MAIL> SEND
To:     MX%BOB
Subj:   ....
```

MX aliases are stored, by default, in a file called MX_ALIAS_ DATABASE.DAT in your login directory (SYS$LOGIN:). You can define the MX_ALIAS_DATABASE logical in your LOGIN.COM to relocate the database file:

```
$ DEFINE MX_ALIAS_DATABASE dev:[user.MAIL]ALIASES.DAT
```

MXALIAS will automatically create the MX alias database the first time you add an alias definition.

MXALIAS can be executed by setting up a foreign symbol in your LOGIN.COM:

```
$ mxalias :== $mx_exe:mxalias.exe
```

Your system manager may have already defined it for you in the system login procedure. You can also just use RUN MX_EXE:MXALIAS to run MXALIAS.

When MXALIAS is invoked without any parameters on the DCL command, your are put into an interactive mode. The prompt is ‘‘MXalias>’’:

```
$ mxalias
MXalias>
```

At the MXALIAS prompt, you can ADD aliases, MODIFY them, REMOVE them, and list them using the DIRECTORY command. There is on-line help available by typing HELP at the MXalias> prompt.

2.1 Adding an MX Alias

The MXALIAS command ADD is used to add an alias to the database. ADD takes three parameters: the alias to define, the equivalent address, and an optional description for the alias. The following example shows a typical definition:

```
MXalias> add joe "smith@somewhere.com" "Joe Smith, Somewhere, Inc."
Added alias JOE to MX alias database
MXalias>
```

The alias, JOE in the example above, can be a string of up to 20 alphanumeric characters (plus $, -, _, and .) that is equated with the given e-mail address. The alias is the address given to MX from the VMS Mail ‘‘To:’’ line using a format like MX%alias. All aliases are converted to uppercase.

The given address must be a valid address in the form ‘‘user@host’’. If the domain is omitted, it defaults to the local host (as defined by the MX_VMSMAIL_LOCALHOST logical). The maximum length of the address is 255 characters. If you want to preserve the case of an address, or if the address contains the ‘‘!’’ character, you must enclose the address in double-quotes. If the address includes quotes, the address should be quoted, with the inside quotes doubled (""""node::user""@domain").

The description is any quoted string of up to 255 characters. The description is displayed by the DIRECTORY command; it is not included in the mail headers of the outgoing message.

2.2 Using an MX Alias

Once an MX alias has been added to the MX alias database, it can be used on the VMS Mail ‘‘To:’’ line by simply prefixing the alias name with MX%. MX will check every address that does not include the ‘‘@’’ character to see if it is an MX alias. For example, if JOE is defined as an alias, the following ‘‘To:’’ line would be specified:

```
MAIL> SEND
To:     MX%JOE
Subj:   ....
```

Sending to MX%‘‘JOE@localhost’’ will prevent MX from performing the alias translation, in case you want to send mail to a local user name JOE.

2.2.1 Disabling System-Wide MX Alias Lookups

MX supports both a personal MX alias database and a system-wide (global) alias database created by your system manager. If an alias is not found in your personal database, MX will see if it’s defined in the global database and, if so, it will use that definition. If your system manager has defined a global database and you wish to prevent MX from accessing it, you can do so by using the following command:

```
$ define mx_global_alias_database _nl:
```

That command causes MX to attempt to open the null device as the global database, which effectively disables a system-wide alias database.

2.2.2 Displaying MX Address Translations

To see the resulting addresses used by MX for all MX% addresses, define the logical MX_VMSMAIL_SHOW_ADDR as TRUE:

```
$ define mx_vmsmail_show_addr true
$ mail

MAIL> SEND
To:     MX%JOE, MX%"MX-List@WKUVX1.WKU.EDU", SYSTEM
  MX rewrote alias JOE as <SYSTEM@WKUVX1.WKU.EDU>
  MX rewrote MX-List@WKUVX1.WKU.EDU as <MX-List@WKUVX1.WKU.EDU>
Subj:   ....
```

The MX_VMSMAIL_SHOW_ADDR works regardless of whether or not MX aliases are specified. If you always want to see MX address translations, you can put the DEFINE command in your LOGIN.COM.

2.3 Displaying Aliases

The MXALIAS command DIRECTORY is used to display your defined aliases. By default, the brief directory listing shows only the alias and the comment, if there is one:

```
MXalias> dir

MX Alias              Description
------------          -----------
JOE                   Joe Smith, Somewhere, Inc.

MXalias>
```

Wildcards can be given to limit the display to aliases matching the given pattern. The DIRECTORY/FULL command can be used to show the equivalent e-mail addresses.

The /OUTPUT=file qualifier can be used to write the directory listing to a text file.

2.4 Modifying Aliases

The MODIFY command is used to modify an existing alias definition. It accepts the alias name as a parameter and the qualifiers /ADDRESS and /DESCRIPTION. For example:

```
MXalias> MODIFY JOE/DESCRIPTION="Local system manager"
Modified alias JOE
MXalias>
```

2.5 Removing Aliases

The REMOVE command is used to remove an alias definition from the MX alias database. By default, it prompts the user for confirmation before removing the specified alias:

```
MXalias> remove joe
Remove JOE <SYSTEM@WKUVX1.WKU.EDU> [N]? y
Removed alias JOE
MXalias>
```

You can supply the qualifier /NOCONFIRM to override the confirmation prompt.

3 Electronic Mailing Lists

When talking about electronic mail, the term *mailing list* is generally used to describe an E-mail address that forwards messages to more than one user. Mailing lists abound on the Internet, on a wide variety of technical and non-technical topics.

Unfortunately, there are no widely-enforced standards on the implementation of mailing lists, so their use will vary depending on the systems on which the mailing lists are set up.

3.1 Typical Internet Mailing Lists

For most Internet mailing lists, there are generally two addresses: one for the mailing list itself, and one for ''administrivia'' (subscription requests, etc.). The administrative address is usually the mailing list name with ''-request'' added. For example, the mailing list for discussing Message Exchange is **MX-List@WKUVX1.WKU.EDU**. Subscription requests, removals, or comments about the list are sent to **MX-List-request@WKUVX1.WKU.EDU**.

Many Internet mailing lists are managed manually, so mail sent to -request addresses can usually be free-form. However, some systems, MX included, have mailing list handlers which process some types of requests automatically, without human intervention. The syntax of the commands you send to these automated handlers will vary from system to system. For example, the MX mailing list processor accepts the following commands:

SUBSCRIBE	for getting added to the list
SIGNOFF	for getting removed from the list
REVIEW	for getting a list of the subscribers
HELP	for getting a help message
QUERY	for getting the status of your subscriber entry
SET [NO]CONCEAL	for concealing or revealing your e-mail address in REVIEW lists
SET [NO]DIGEST	for switching between reception of regular list traffic and periodic digests (only applies to lists that have corresponding digests)
SET [NO]MAIL	for enabling/disabling receipt of list messages while remaining subscribed to the list
SET [NO]REPRO	for enabling/disabling receipt of your own list postings
QUIT	for preventing the parsing of a mail signature

Commands must generally be placed in the body of a mail message, rather than on the Subject line.

4 Mail-Based File Servers

The term *file server*, for the purposes of this document, refers to a network entity that maintains a library of files and delivers them to users on demand. While most files are served via FTP or HTTP, some older systems may still use e-mail for file delivery.

There are no standards for mail-based file servers. There are several file server implementations in existence: LISTSERV, VMSSERV, MAILSERV, and several others. MX also includes a file server module, generally referred to as FileServ.

4.1 Get HELP

If you want to obtain files from a file server, and you are unsure of the commands you need to use, you should begin by requesting help information from the server. The best way to do this is to send an E-mail message to the file server's address with the word HELP on the subject line *and* on the first and only line of the body of the message. Most servers will mail you back a message listing the commands they accept and the format the commands should take, along with other helpful information.

If you cannot get assistance from the file server itself, you may be able to get some from the postmaster on the file server's system.

4.2 MX FileServ Commands

The MX file server, usually called FileServ, accepts commands, one command per line, in the body of an E-mail message. The commands it accepts are:

ADDRESS valid-address	provides a valid e-mail address
LIST [pattern]	lists all packages matching ‘‘pattern’’
DIRECTORY [pattern]	same as LIST
SENDME package[.part]	sends an entire package or the specified part
HELP	sends a help message
QUIT	causes any lines following this command to be ignored

FileServ commands may be abbreviated to their shortest unique string.

ADDRESS provides the user with the ability to specify a valid RFC822-compliant e-mail address to which any FileServ output is to be sent. Normally, any files requested from FileServ are sent to the address in the ‘‘Reply-To:’’ or ‘‘From:’’ lines in the message headers. However, addresses are sometimes corrupted by gateways through which the message passes, resulting in an invalid return address. File server users can use the ADDRESS command to provide a valid alternate to the ‘‘From:’’ address.

Note: When an ADDRESS command is processed, the file server transaction log includes the original ''From:'' address. Any user receiving unasked-for files can use it to determine from whom the request came.

4.2.1 Packages

A *package* is a collection of related files that are grouped together for distribution. FileServ, along with other file servers, distributes files in packages. These packages are usually in a special format for distribution over the network via E-mail; once you collect all of the parts in a package, the parts are combined together and fed through an unpacking program (sometimes contained within the package itself) to recreate the original collection of files.

4.2.2 Binary Files

Because E-mail systems generally do not properly handle binary data, binary files (such as executable images or compressed files) are generally *encoded* before being packaged and distributed by a file server. Once unloaded from the package, the encoded file must then be decoded to recreate the binary file. The type of encoding will vary from system to system.

In addition, large files may be *compressed* before being encoded and packaged, to cut down on the network bandwidth required when transmitting the package. Restoring the original files then requires an additional decompression program.

A Message Header Format

Most network mail systems require or include more information about messages than VMS MAIL can handle. MX, for example, follows the Internet message format standard, usually called *RFC 822* after the number of the document that describes the format.

When you receive a message via MX, the FROM address identified in the VMS MAIL headers will begin with the MX% prefix, which allows you to REPLY to the message. In addition to the VMS MAIL headers, you will also see the RFC 822 header information, which is usually displayed as the first part of the message text (this is under the control of the system manager). For example:

```
    #1          29-FEB-1992 10:36:22.11                        NEWMAIL
From:   MX%"naive@myhost.mycompany.com"
To:     MANAGER
CC:
Subj:   Question

Return-Path: <naive@myhost.mycompany.com>
Received: from myhost.mycompany.com by mgrsta.mycompany.com (MX V3.0);
          Thu, 29 Feb 1992 10:35:10 EST
Received: by myhost.mycompany.com (MX V3.0) id 31437; Thu, 29 Feb 1992
          10:35:05 EST
Resent-Date: Thu, 29 Feb 1992 10:35:01 EST
Resent-From: system@myhost.mycompany.com
Resent-To: manager@mgrsta.mycompany.com
Sender: <naive@myhost.mycompany.com>
Date: Thu, 29 Feb 1992 10:34:55 EST
From: Naive User <naive@myhost.mycompany.com>
Reply-To: naive@myhost.mycompany.com
Message-ID: <00933068.08a17f00.31437@myhost.mycompany.com>
To: system@myhost.mycompany.com
Subject: Question

How do I send E-mail?
```

The first five lines of this message are the VMS MAIL headers. The message text starts with the RFC 822 headers, followed by the message itself. The following sections explain the meaning of the RFC 822 headers.

Return-Path. The return address as appears on the envelope of the message. This usually identifies the route the message took in getting to you, and can be used to identify forged messages in some cases. The return path is used as the VMS MAIL From address if no other address is available.

Received. There may be several of these lines for a message. They usually indicate how and when the message was transferred from one system to another. They are provided for informational purposes only.

Resent- lines. If the message is forwarded (usually by an automatic mechanism such as SET FORWARD in VMS MAIL), some messaging systems (MX included) will include information about when it was forwarded and who it was forwarded to. One set of Resent lines usually appears for each forwarding hop.

Sender. This line indicates the sender of the message, which could be different from the address in the From line.

Date. This line indicates the date and time the message was entered into the mail system by the sender. It will usually include the local time for the sender, which may be in a different time zone.

From. This line indicates who the message is from. If the message was sent by someone on behalf of another person or group, the message will also include a *Sender* line to identify the person or agent who actually sent the message.

Reply-To. If the sender wants to receive replies at an address different from the From address, a Reply-To line will be included to redirect the replies.

Message-ID. The message identifier uniquely identifies a message. Message-ID's are used by some mail systems for tracking messages and replies.

To. Identifies the target user or users for the message. Also included can be **CC** and **BCC** lines that identify users to whom a carbon copy and ''blind'' carbon copy of the message is sent.

Subject. A brief description of the subject of the message.

Other headers are also possible, some of which are extensions to the RFC 822 message standard. Also, the order in which the headers appear may vary from system to system.

A.1 VMS MAIL Headers

MX automatically translates some of the RFC 822 header information into VMS MAIL headers.

A.1.1 From Header

There are several RFC 822 headers used for identifying the originator of a message. VMS MAIL, however, allows only one. To allow the REPLY command to work properly, therefore, MX fills in the VMS MAIL From line with the address that should be used in generating a reply. This reply address is selected from one of the following header lines, listed here in order of preference:

1 Reply-To

2 From

3 Sender

4 Return-Path

MX will only use the address from one of these headers if it is syntactically valid. Since most mail systems provide a valid address in the Reply-To and/or From headers, this should not be a problem.

A.1.2 To and CC Headers

The VMS MAIL To and CC headers will list only the users on the local system receiving the message. To see the actual list of recipients, examine the To, CC, and BCC lines in the RFC 822 headers.

A.1.3 Subject Header

The VMS MAIL Subject header should be identical to the RFC 822 Subject header, if one exists.