Message Exchange Management Guide

August, 2008

This manual describes the management and operation 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 ix

CHAPTER 1 OVERVIEW OF MESSAGE EXCHANGE OPERATION 1–1

1.1 WHAT IS A MESSAGE? 1–1

1.2 WHAT IS AN ADDRESS? 1–2

1.3 MX COMPONENTS 1–2
1.3.1 The Message Queue 1–3
1.3.2 Message Entry Agents 1–3
1.3.3 The Router 1–3
1.3.4 Delivery Agents 1–4
1.3.5 MLF Agent 1–4

CHAPTER 2 CONFIGURING MX WITH MXCONFIG 2–1

2.1 WHY USE MXCONFIG? 2–1

2.2 USING MXCONFIG 2–1

CHAPTER 3 MANAGING THE ROUTER 3–1

3.1 REWRITE RULES 3–1
3.1.1 Regular Expressions in Rewrite Rules 3–1

3.2 DEFINING DELIVERY PATHS 3–2

3.3 ALIAS TRANSLATION 3–2

3.4 CONTROLLING THE ROUTER PROCESS 3–3

3.5 LOGGING ROUTER EVENTS 3–3

CHAPTER 4 MANAGING THE DELIVERY AGENTS 4–1

4.1 LOCAL DELIVERY OPTIONS 4–1

4.2 SMTP AND DECNET_SMTP DELIVERY OPTIONS 4–1
4.2.1 Internet "Mail Exchanger" Support 4–1
4.2.2 Character Set Conversion 4–2
4.2.3 Default SMTP Router 4–2
4.2.4 SMTP Lock Files 4–2
4.2.4.1 Excluding Hosts from Locks • 4–3

4.3 SITE DELIVERY OPTIONS 4–3

4.4 SHUTDOWNS AND RESETS 4–3

4.5 LOGGING DELIVERY AGENT EVENTS 4–4

CHAPTER 5 MANAGING MESSAGE ENTRY AGENTS 5–1

5.1 LOCAL MESSAGE ENTRY 5–1
5.1.1 System-wide MX Aliases for Outgoing Addresses 5–1
5.1.2 VMS MAIL Protocol Prefix 5–2
5.1.3 Default Host Names 5–2
5.1.4 Character Set Conversion 5–3

5.2 SMTP_SERVER 5–3
5.2.1 SMTP Server Threads 5–3
5.2.2 Limiting "Outside" Thread Usage 5–3
5.2.3 Setting the SMTP Listener Port 5–4

5.3 DECNET_SMTP NETWORK OBJECT 5–4

5.4 MESSAGE ENTRY AGENT SHUTDOWNS 5–5

CHAPTER 6 MANAGING THE MESSAGE QUEUE 6–1

6.1 ESTABLISHING THE QUEUE SIZE 6–1

6.2 RUNNING THE MX FLQ MANAGER 6–1

6.3 QUEUE CLEANUP LOGICALS 6–2

6.4 AUTOMATIC PURGING OF FINISHED QUEUE ENTRIES 6–2

6.5 THE MCP QUEUE COMMANDS 6–2
6.5.1 Interpreting MCP QUEUE SHOW Output 6–3
6.5.2 Forcing a Message to "Bounce" 6–4
6.5.3 Interpreting MCP QUEUE STATISTICS Output 6–4

CHAPTER 7 CONFIGURING MX FOR PART-TIME INTERNET CONNECTIONS 7–1

7.1 ENVIRONMENT FOR PART-TIME CONNECTIONS 7–1

7.2 CONFIGURING MX AS A PART-TIME CLIENT 7–1

7.3 CONFIGURING MX AS A SERVER FOR PART-TIME CLIENTS 7–2

7.4 ETRN SUPPORT 7–2

7.5 EXAMPLE CONFIGURATION 7–3

CHAPTER 8 REDUCING OR ELIMINATING "JUNK" E-MAIL 8–1

8.1 WHAT IS "JUNK" E-MAIL? 8–1

8.2 MX FILTERING FEATURES 8–1

Contents

8.3 SENDER VALIDATION 8–2

8.4 DISABLING THE SMTP RELAY FUNCTION 8–2
8.4.1 Identifying Inside Networks and Hosts 8–3
8.4.2 Disabling the "Percent Hack" 8–4

8.5 REALTIME BLACKHOLE LIST 8–4

8.6 ESTABLISHING REJECTION CRITERIA WITH REJMAN 8–5
8.6.1 Address-Based Rejections 8–5
8.6.2 Header-based Rejections 8–6
8.6.3 Tracking Rejections 8–7
8.6.4 Purging Old Rules 8–7

8.7 HEURISTIC FILTERS 8–8
8.7.1 Confidence Levels 8–8
8.7.2 Rejection Actions 8–10
8.7.3 Junk Mail Warnings 8–11

8.8 LOGGING SMTP SERVER REJECTIONS 8–11

8.9 DEBUGGING REJECTION RULES AND HEURISTIC FILTERS 8–11

8.10 REDUCING JUNK MAIL POSTINGS ON MAILING LISTS 8–12
8.10.1 Ignoring Heuristically-Determined Junk Mail 8–12
8.10.2 Ignoring Messages Without the List Address 8–12

8.11 USING SMTP AUTHENTICATION TO BYPASS JUNK MAIL CHECKING 8–12
8.11.1 Authentication Mechanisms Supported 8–13
8.11.2 Managing the SMTP Authentication Database 8–13
8.11.3 How Authentication Works 8–14
8.11.4 PLAIN Authentication Features 8–15
8.11.5 Limiting SMTP Authentication Access to Certain Users 8–15
8.11.6 Intrusion Detection and Evasion 8–15
8.11.6.1 Authentication Retry Timers • 8–15
8.11.6.2 Authentication Retry Limits • 8–15
8.11.6.3 Intrusion Evasion • 8–16

CHAPTER 9 OTHER MISCELLANEOUS UTILITIES 9–1

9.1 THE MLFAKE UTILITY 9–1

9.2 THE MAILQUEUE UTILITY 9–2

9.3 THE MX_DECODE UTILITY 9–2

CHAPTER 10 TROUBLESHOOTING MX 10–1

10.1 QUEUE FILES USED BY MX COMPONENTS 10–1
10.1.1 File Types 10–1

10.2 PROCESS NAMES 10–2

10.3 DEBUG/TRACE OUTPUT 10–2

CHAPTER 11 THE MX STARTUP PROCESS 11–1

11.1 STARTUP COMMAND PROCEDURES 11–1

11.2 STARTUP DATA FILES 11–1

MCP COMMAND DICTIONARY

MCP MCP–3
@ (REDIRECT COMMAND INPUT) MCP–4
ADD USER MCP–5
ATTACH MCP–6
CREATE USER_DATABASE_FILE MCP–7
DEFINE/KEY MCP–8
DEFINE ALIAS MCP–9
DEFINE FILE_SERVER MCP–10
DEFINE INSIDE_NETWORK_ADDRESS MCP–13
DEFINE LIST MCP–15
DEFINE LOCAL_DOMAIN MCP–23
DEFINE PATH MCP–24

DEFINE REWRITE_RULE MCP–25
DEFINE SYSTEM_USERS MCP–27
EXIT MCP–28
HELP MCP–29
MODIFY MCP–30
MODIFY USER MCP–31
QUEUE CANCEL MCP–32
QUEUE COMPRESS MCP–33
QUEUE CREATE MCP–34
QUEUE DUMP MCP–35
QUEUE EXTEND MCP–37
QUEUE HOLD MCP–38
QUEUE PURGE MCP–39
QUEUE READY MCP–40
QUEUE SELECT MCP–41
QUEUE SHOW MCP–43
QUEUE STATISTICS MCP–46
QUEUE SYNCHRONIZE MCP–47
QUIT MCP–48
REMOVE MCP–49
REMOVE USER MCP–50
RESET MCP–51
REVIEW MCP–52
SAVE MCP–53
SET DECNET_SMTP MCP–54
SET LOCAL MCP–55
SET MLF MCP–58
SET ROUTER MCP–59
SET SITE MCP–61
SET SMTP MCP–62
SHOW MCP–65
SHUTDOWN MCP–67
SPAWN MCP–68
STATUS MCP–69

REJMAN COMMAND DICTIONARY

REJMAN REJMAN–3
COMMAND INPUT REDIRECTION REJMAN–4
ADD EXCLUSION REJMAN–5
ADD REJECTION REJMAN–6
ATTACH REJMAN–9
CHECK REJMAN–10
DEFINE/KEY REJMAN–11
DELETE EXCLUSION REJMAN–12
DELETE REJECTION REJMAN–13
DISABLE HEURISTIC REJMAN–14
ENABLE HEURISTIC REJMAN–15
EXIT REJMAN–17
HELP REJMAN–18
QUIT REJMAN–19

PURGE REJMAN–20
SAVE REJMAN–21
SET HEURISTICS REJMAN–22
SHOW REJMAN–24
SPAWN REJMAN–25

APPENDICES

APPENDIX A REGULAR EXPRESSIONS A–1

A.1 SPECIFYING REGULAR EXPRESSIONS A–1
A.1.1 Examples A–3
A.1.2 Substitutions A–3

INDEX

FIGURES

1–1 Message parts 1–1

TABLES

5–1 Address Formatting Logical Names 5–2
6–1 FLQ Manager/Router queue-related logicals 6–2
8–1 Heuristic Filters 8–9
8–2 Intrusion Detection and Evasion Parameters 8–16
10–1 Debug/Trace logical names 10–3
11–1 Component names for use with MX_STARTUP.COM 11–1
MCP–1 Owner Notification Types MCP–18
MCP–2 Mailing list protection classes MCP–19
MCP–3 Mailing list protection codes MCP–19
MCP–4 Typical protection codes MCP–19
MCP–5 QUEUE DUMP Output Files MCP–35
MCP–6 Header name keywords MCP–56
MCP–7 MCP STATUS Descriptions MCP–69
A–1 Atom Sequencing Terms A–1
A–2 Atom Sequences A–1
A–3 Collating Elements in Bracket Expressions A–2
A–4 Character Classes A–2

Preface

This guide describes the management and operation of Message Exchange (MX).

Intended Audience

This manual is intended for use by the system manager or any individual responsible for installing and maintaining MX. The reader should be generally familiar with VMS system concepts, electronic mail systems and networking terminology.

Document Structure

This guide consists of two parts. Part I contains nine chapters which contain information on management and operation of the various components of MX. Part II is the command dictionary for the MX Control Program (MCP).

Chapter 1	Contains information about how Message Exchange operates.
Chapter 2	Describes how to use the MXCONFIG procedure.
Chapter 3	Describes how to manage the Router functions.
Chapter 4	Describes how to manage the message delivery agents.
Chapter 5	Describes how to manage the message entry agents.
Chapter 6	Describes how to manage the message queue.
Chapter 7	Describes some miscellaneous MX utilities.
Chapter 8	Describes the tools available for troubleshooting MX.
Chapter 9	Describes the MX startup process.

Related Documents

You can find additional information in the following documents:

- *Message Exchange Installation Guide* describes the installation of MX.
- *Message Exchange User's Guide* describes MX features available to general users.
- *Message Exchange Programmer's Guide* describes the programmable customization features
- *Message Exchange Mailing List/File Server Guide* describes the MX Mailing List/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.
- *RFC 821: Simple Mail Transfer Protocol* describes the SMTP protocol.

- *RFC 822: Standard for the Format of ARPA Internet Text Messages* describes the format of headers and addresses used by Internet hosts.
- *RFC 1123: Requirements for Internet Hosts – Application and Support* provides additional information on SMTP support for Internet hosts.

1 Overview of Message Exchange Operation

This chapter briefly describes how MX operates.

1.1 What is a Message?

Electronic mail messages are usually divided up into three parts:

- The *envelope*. Much like an envelope used for mail in the real world, an electronic mail envelope includes a return address and destination information. Unlike real mail, however, one message can have multiple destinations. In addition, addresses on the envelope can be changed as they pass through a system.
- The *headers*. Message headers include information about the message that the recipient will see when he or she reads the message. This information includes the date the message was sent, the subject of the message, who sent it and who will receive it, and which systems the message passed through on its way to the recipient.
- The *body*. This is the message text itself, as entered by the person (or other entity) that sent the message.

There are several standards for the format of each part of a message. MX uses the Internet RFC 822 format for message headers and body, and Internet RFC 821 format for envelope information. When sending messages to non-Internet sites, MX will convert the message format as needed to comply with the standards required by the destination system. Figure 1–1 is an example of a message broken down into its parts.

Figure 1–1 Message parts

```
Envelope:

    <user1@host1.org>       Return address
    <user2@host2.org>       Recipient #1
    <user3@host3.org>       Recipient #2

Headers:

    Received: from host1.org by host2.org with SMTP; 01 Oct 1990 12:32:01 EDT
    Date: Mon, 01 Oct 1990 11:19:47 EDT
    From: user1@host1.org
    To: user2@host2.org
    Cc: user3@host3.org
    Subject: Hello there

Body:

    Just a quick note to let you know I'm alive.
    Have a nice day.
```

1.2 What is an Address?

Much like the address on a real envelope, an electronic mail address indicates where a message should be delivered, or where it came from. MX uses the Internet RFC 822 format for addresses. RFC 822 specifies a very rich syntax for addresses, but most are of the form:

local-part@domain

Where **domain** usually identifies a system and **local-part** identifies the user on that system.

Envelope Addresses

Envelope addresses are kept by MX in a special format, the route-address, which adheres to Internet RFC 821. Users cannot generally use route-addresses when addressing mail; they are used internally by MX and other mail systems for tracking the route a message has taken to get from source to destination, or for forcing a particular route to be taken for a message.

A route-address has the form

<local-part@domain>
or
<@domain[,@domain...]:local-part@domain>

This form of addressing is discouraged on the Internet, but is used when messages are gatewayed between multiple mail networks.

1.3 MX Components

Message Exchange consists of several parts:

- A *message queue*, where all messages are stored during processing by MX.
- *Message entry agents*. These programs or processes take messages in from users or from other networked hosts and enter them in the message queue for processing.
- The *Router*. This is the ‘‘hub’’ of MX processing. All incoming messages have their envelope information processed by the Router to determine how they should be delivered.
- *Message delivery agents*. These programs or processes take messages that have been processed by the router and deliver them either to local users or to other networked hosts.
- The *Mailing List/File Server* (MLF) agent. This special process handles all mailing list and file server requests.

1.3.1 The Message Queue

All MX messages are stored in a directory called the message queue (sometimes called the *file queue*). This is the directory pointed to by the logical name MX_FLQ_DIR. Besides the files comprising the messages themselves, the queue directory also contains a file called MX_SYSTEM_QUEUE.FLQ_CTL. This file, called the *queue control file*, is a sequential file that contains information about the state of each message, who is processing it, etc. All MX processes access their queue entries through this control file.

The size of the queue control file determines the maximum number of entries that can be in the queue at any given time. The larger the file, the more entries that can be in the queue.

Because the message queue is shareable cluster-wide, a user on any node in a VMScluster can send messages over a network, even if there is no direct network connection (via TCP/IP or DECnet) on the particular node to the target network.[1]

1.3.2 Message Entry Agents

Messages are entered into MX by users from VMS Mail through the MX% protocol prefix. This invokes routines in image MX_EXE:MX_MAILSHR.EXE, which create the necessary files in the message queue for processing by the Router.

Messages coming in from other hosts are handled by

- an SMTP server, for messages coming in over TCP/IP;
- a DECnet-SMTP server, for messages coming in via SMTP-over-DECnet;
- the MX_SITE_IN program, for messages coming in from a locally-created network interface.

Messages are also entered into the queue by the Mailing List/File Server (MLF) agent, in response to a mailing list or file server request.

1.3.3 The Router

The Router is responsible for taking the envelope information from a message and determining where the message should be sent based on the addresses listed in the envelope.

Each recipient address in the envelope is processed in two or three phases:

1 In the *rewrite* phase, the address is checked against a list of rewriting rules. If it matches one of the rules, the rule is applied and the original address is replaced.

[1] When following the MX clustering guidelines described in *Message Exchange Installation Guide*.

2 In the *path identification* phase, the *next hop* domain of the address is identified and that domain is checked against the domain-path mapping list. This identifies the delivery agent that will be called on to deliver the message to the recipient.

3 If the recipient is on the local system, a third phase is entered, which checks to see if the *local-part* of the address is an alias for another address, a mailing list name, or file server name.

The Router is also responsible for maintaining the message queue. It cleans out completed or cancelled entries.

1.3.4 Delivery Agents

The Local delivery agent delivers mail to local users or to other hosts over DECnet using VMS Mail. It also identifies local users who have used SET FORWARD to direct their mail elsewhere and resends messages to their forwarding addresses.

Other delivery agents send messages to other hosts or other mail-processing software.

- The SMTP delivery agent sends messages using the Simple Mail Transfer Protocol over TCP/IP.
- The DECNET_SMTP delivery agent sends messages using the Simple Mail Transfer Protocol over DECnet.
- The SITE delivery agent passes messages to a locally-created network interface package.

Each delivery agent is responsible for converting MX-format messages into the format required for the particular network or network interface package.

1.3.5 MLF Agent

The Mailing List/File Server (MLF) agent is a special form of delivery agent that handles mailing list and file server requests. It doesn't actually deliver messages to a network directly. What it does is create new messages based on the list or server requests and sends the new messages back to the Router for processing and eventual delivery.

2 Configuring MX with MXCONFIG

This chapter describes the MXCONFIG procedure, MX_DIR:MXCONFIG.COM.

2.1 Why Use MXCONFIG?

Configuring MX by hand can be a complicated and error-prone process, due to the number of options available. Based on a question-and-answer script, MXCONFIG creates the MX startup information for its logical names and agent invocation, as well as a command file that will generate an MX configuration database. Configurations created with MXCONFIG should be adequate for most Internet sites; it can also be used as a base that can be tailored using the MX Control Program (MCP), if needed.

2.2 Using MXCONFIG

When you execute MXCONFIG, it displays a menu of configuration options:

```
                Message Exchange Configuration Procedure

Main Menu

    1. Configure MX message queue logical names.
    2. Configure MX host and timezone logical names.
    3. Configure MX agent processes.
    4. Create an MX configuration database.
    5. Exit from this procedure.

Enter choice:
```

To completely reconfigure MX, start with menu option 1, then select each of the other options in order as you complete each section. Each section displays information about the items being configured, then asks you to answer some questions. Read the explanations carefully before answering.

When you are finished with your configuration changes, select option 5 from the main menu to exit from the configuration procedure. You will be asked whether you would like to save the configuration changes before the procedure exits.

3 Managing the Router

This chapter describes the MCP commands used to configure and control the Router.

3.1 Rewrite Rules

Address-rewriting rules, or *rewrite rules* for short, are checked by the Router for every recipient address on every envelope of every message that passes through MX. A rewrite rule consists of a pattern and a result. If an address matches the pattern, the rule is applied and the address rewritten per the rule's result. The purpose of this is to provide a general means of altering envelope addresses, primarily for handling multi-gateway cases where DEFINE PATH/ROUTE is insufficient.

Be careful, since the rule processor treats the addresses as ordinary text strings and does not understand the syntax of RFC 821 addresses. Because they were designed mainly for handling domain aliases, rewrite patterns are matched from right to left.

The rewrite rule list is searched only once per address, until a matching pattern is found. Once a match is found, no additional rules are searched. If no rule matches an address, further processing continues on the original address.

An example of an application for rewrite rules is the mapping of an artificial domain name, such as *host***.dnet**, into an address for delivery through VMS MAIL over DECnet:

```
MCP> DEFINE REWRITE_RULE "<{user}@{host}.dnet>" -
_MCP>                    "<""{host}::{user}""@local.host.name>"
```

The pattern matching routine treats the variable references in the first string as wildcards; everything between the left angle bracket and the at sign is copied into the {user} variable, and everything between the at sign and the string **.dnet>** is copied into the {host} variable. The variable names have no special significance to the pattern matching routine.

3.1.1 Regular Expressions in Rewrite Rules

For more sophisticated pattern matching, MX supports the use of UNIX-style regular expressions in rewrite rules. Specify the /REGEX qualifier on the DEFINE REWRITE_RULE command to have the rule invoke the regular expression parser. The example rule show above could be specified as:

```
MCP> DEFINE REWRITE_RULE/REGEX "<([[:username:]]+)@([[:domain:]]+)\.dnet>" -
_MCP>                          "<""\2::\1""@local.host.name>"
```

When using regular expressions, use parentheses to enclose subexpressions that you would like to include in the rewritten address. Up to 9 such subexpressions can be used; to instantiate them in the rewritten address, use a backslash followed by a digit (1-9).

MX's regular expression support includes two character classes, ''[:username:]'' and ''[:domain:]'', in addition to the standard classes. These additional classes match those characters permitted in the username portion and domain name portion of SMTP addresses, respectively. See Appendix A for further information on regular expressions.

3.2 Defining Delivery Paths

The first step the Router takes in determining a delivery path is to identify the next hop the message should take. The next hop is determined by looking at the address and selecting either the first domain in the route path at the beginning of the address, or if there is no route path, the destination domain. The second step is to search the list of defined domain/path mappings to determine the delivery path, and possibly a routing host for that domain.

The MCP DEFINE PATH command is used to create a domain/path mapping. A mapping consists of a domain pattern (possibly containing VMS wildcard characters) and the name of the delivery path to be used if the next hop matches the domain pattern. Possible paths are DECNET_SMTP, HOLDING_QUEUE=*n*, LOCAL, SITE, and SMTP.

For example, a typical path list for an Internet host might be created with the commands:

```
MCP> DEFINE PATH myhost.mycompany.ORG   LOCAL
MCP> DEFINE PATH myhost                 LOCAL  ! abbreviation
MCP> DEFINE PATH [1.2.3.4]              LOCAL  ! numeric address
MCP> DEFINE PATH *                      SMTP
```

The path list is searched sequentially until a match is made. The first three rules catch any locally-addressed messages. The next two rules provide transparent routing of addresses in the UUCP ''fake domain'' through an Internet gateway. The last rule, which would match any other domain name, routes all other messages off-system via SMTP. Notice that abbreviations or nicknames for the local host must have LOCAL path definitions to be recognized by MX.

3.3 Alias Translation

The third phase of Router address processing is the identification and translation of local aliases. The system manager or postmaster can define aliases on the local system that translate to any local or remote address with the MCP DEFINE ALIAS command. If an address, after passing through the first two Router phases, is identified as a local address, the Router searches the alias list. If the local part of the original address matches one of the aliases, the original address is discarded and the alias address is substituted in its place and is passed through the other address processing phases.

Note that alias processing is totally transparent to the sender as well as the recipient of a message. No message headers are changed or added to indicate that the message is being forwarded via an alias address. In addition, aliases are kept in a simple list that is searched sequentially, rather than a more efficient structure. For these two reasons, it is recommended that aliases be used sparingly. Mail forwarding is better done with the VMS MAIL SET FORWARD command.

Also performed during this phase is ‘‘percent-dehacking’’ of addresses. MX supports the ‘‘percent-sign hack’’ that allows users to route messages through the local system by specifying an address of the form ‘‘user%host1@host2’’. If the local part of the address is found to contain a percent sign, the percent sign is converted to an at sign, the original address is discarded, and the new address is substituted as for aliases. While this form of routed addressing is not recommended, it is sometimes required when the local host is acting as a gateway between two networks. You can disable the percent-dehacking function with the MCP command SET ROUTER/NOPERCENT_HACK.

3.4 Controlling the Router Process

The Router process will respond to shutdown and reset signals sent by the MCP SHUTDOWN and RESET commands, respectively. Using these commands is the only way that the Router can be shut down or reset without possibly losing messages.

You can control Router functions, such as creation of an accounting log, with the MCP SET ROUTER command.

3.5 Logging Router Events

Major events in the Router process, such as startup, shutdown, and configuration resets, are automatically logged to the Router’s log file, MX_ROUTER_DIR:MX_ROUTER_nodename.LOG_*process-id*. These events may also be logged to an operator console by defining the logical name MX_EVENT_OPER_CLASS:

```
$ DEFINE/SYSTEM/EXEC MX_EVENT_OPER_CLASS class-name
```

where *class-name* can be any recognized OPCOM operator class, such as NETWORK.

This logical name must be defined **before** MX is started in order to have any effect. Its definition affects all MX processing agents.

4 Managing the Delivery Agents

This chapter describes some of the MCP commands used to configure and control the various MX delivery agents.

4.1 Local Delivery Options

The local delivery agent can be configured to place message header lines at either the beginning of the message text, the end of the message text, or both, when delivering locally through VMS Mail.

In addition, you can control whether accounting information is generated, the delivery retry interval, and the maximum retry count. By default, unsuccessful deliveries into VMS Mail are retried every half hour up to 96 times total (giving a two-day period) before being returned to sender.

The MCP SET LOCAL command can be used to alter any of these settings; refer to the command description for further information.

4.2 SMTP and DECNET_SMTP Delivery Options

As with the local delivery agent, you can alter the accounting setting, the retry interval, and the maximum retry count for SMTP and DECNET_SMTP deliveries. However, the SMTP agent differentiates between failed deliveries due to domain name lookup failures and other kinds of failed deliveries, and you can set a different maximum retry count for DNS lookup failures. The MCP SET SMTP and SET DECNET_SMTP commands are used to alter the settings for the three delivery agents. The defaults are 30 minutes for retry interval, 12 DNS failures maximum (for SMTP only), and 96 general failures maximum.

Refer to the command descriptions for further information.

4.2.1 Internet "Mail Exchanger" Support

Some of the supported TCP/IP packages include domain name resolvers that provide access only to host name-to-address mapping information. However, not all Internet domain names map directly to addresses. Domain names are also used to identify hosts on other networks to which electronic mail can be sent via some other Internet-connected gateway host, called a *mail exchanger*.

Mail exchangers are recorded in the Internet Domain Name System (DNS) using mail exchanger resource records (MXRRs). The initial list of DNS servers to be asked for MXRR information is controlled by the NETLIB software. Refer to the NETLIB documentation for further information.

4.2.2 Character Set Conversion

The Local Delivery agent performs character conversion when formatting the VMS MAIL From: and Subject: lines from the contents of the RFC822 message headers, and can also perform conversion of non-MIME message bodies, or MIME messages with a Content-Type of text/plain. This conversion is the inverse of the conversion performed by the VMS MAIL interface; see Section 5.1.4 for more information.

4.2.3 Default SMTP Router

When the local system uses host tables instead of Domain Name Service, you may want to establish a default router for SMTP messages. The SMTP delivery agent will automatically forward to the default router all messages addressed to users on hosts whose names are not found either in the Domain Name System or in the local host table provided by your TCP/IP package.

A default router is established in MCP with the SET SMTP/DEFAULT_ ROUTER command.

Before you use a default router, you should ensure that:

- The host name for the system you are using as a default router is known to your system's TCP/IP (i.e., is in your system's host tables).
- The default router you select ''knows'' more about the Internet than your host, or in turn can forward to another host that has access to more domain name information.
- You have the consent of the people managing the system you intend to use as a default router. This is especially important if you expect the traffic between your system and the default router to be heavy.

4.2.4 SMTP Lock Files

To prevent SMTP delivery agents from getting tied up with delivering messages to sites that are unreachable, they create ''lock files'' to track failures with specific domains and hosts that they have failed to reach. These files are created in MX_SMTP_LOCK_DIR:, which by default is located at MX_ROOT:[SMTP.LOCK].

Before an SMTP agent tries to contact a mail server, it first checks to see if a lock file exists for that server. If so, it compares the time that file was last modified against the SMTP retry interval. If the retry interval has not been exceeded, it immediately returns the message being processed to the queue for a later attempt. If no lock file exists, or the lock file is older than the retry interval, the SMTP agent attempts to contact the mail server. If the server is unreachable, the SMTP agent creates a lock file to prevent other attempts to reach that server until the retry interval has elapsed.

Lock files are used for both domain names as well as the individual IP addresses that the SMTP agent tries to reach for each domain. The lock file directory is automatically cleaned periodically during the message queue cleanup process invoked by either the Router or the FLQ Manager agent.

4.2.4.1 Excluding Hosts from Locks

Occasionally, there may be mail servers that you need to exclude from the lock file checking; for example, a local mail server that your system sends many messages to, but is subject to occasional recahability problems that causes excessive backlogs in the MX message queue. You can exclude these hosts from lock file checks by defining the logical name MX_SMTP_LOCK_EXCLUSIONS (in executive mode). You should define this logical name with both the domain names as well as the IP addresses of the hosts you need to exclude; wildcards are permitted.

For example, if you need to exclude all of the hosts in domain ‘‘mycompany.com’’, all of which are on the IP network ‘‘10.12.100.0’’, you would define the logical name as follows:

```
$  DEFINE/SYSTEM/EXEC MX_SMTP_LOCK_EXCLUSIONS -
_$                    "*.mycompany.com",-
_$                    "10.12.100.*"
```

This will prevent the SMTP agents from locking out any mail server under this domain (and also on this network) from being locked out.

You can prevent the SMTP agents from creating any lock files by defining MX_SMTP_LOCK_EXCLUSIONS as ‘‘*’’. However, this is not recommended.

4.3 SITE Delivery Options

The SITE delivery agent includes support for retry on error. The MCP SET SITE command can be used to alter the retry interval and maximum retry count. Refer to the SET SITE command description for further information.

4.4 Shutdowns and Resets

Each of the delivery agents will respond to shutdown and reset signals as sent by the MCP SHUTDOWN and RESET commands, respectively. Using these commands is the only guaranteed way of cleanly shutting down and resetting the delivery agents, without loss of messages in progress.

The MCP SHUTDOWN command can be forced to wait for the completion of a shutdown by using the /WAIT qualifier. This qualifier is recommended when including an MX shutdown sequence in your system’s SYSHUTDWN.COM procedure.

There may be times when it is necessary to prevent local users from using VMS Mail to send mail via MX. To do so, define the executive-mode system logical name MX_SHUTDOWN:

```
$ DEFINE/SYSTEM/EXEC MX_SHUTDOWN TRUE
```

If a user tries to send mail to an MX% address and MX_SHUTDOWN is defined, VMS Mail (MX_MAILSHR) will display an error message stating that MX has been temporarily disabled by the system manager.

4.5 Logging Delivery Agent Events

Major events in the delivery agents, such as startup, shutdown, and configuration resets, are automatically logged to each agent's log file. These events may also be logged to an operator console by defining the logical name MX_EVENT_OPER_CLASS:

```
$ DEFINE/SYSTEM/EXEC MX_EVENT_OPER_CLASS class-name
```

where *class-name* can be any recognized OPCOM operator class, such as NETWORK.

This logical name must be defined **before** MX is started in order to have any effect. Its definition affects all MX processing agents.

5 Managing Message Entry Agents

This chapter describes the options available with the MX message entry agents.

5.1 Local Message Entry

The VMS MAIL interface (MX_MAILSHR) is used for local message entry. It is controlled through the definition of system-wide logical names.

Use of MX through VMS Mail can be restricted by defining the executive-mode logical MX_RESTRICT_USAGE in the system logical name table:

```
$ DEFINE/SYSTEM/EXEC MX_RESTRICT_USAGE TRUE
```

If the logical is defined, the user must hold the MX_MAIL_ACCESS process rights identifier in order to send mail using MX. The VMS utility AUTHORIZE is used to create and grant identifiers:

```
$ set default sys$system:
$ run authorize
UAF> ADD/IDENTIFIER MX_MAIL_ACCESS
Identifier MX_MAIL_ACCESS value: %X8001000D added to rights data base
UAF> GRANT/IDENTIFIER MX_MAIL_ACCESS GOATHUNTER
Identifier MX_MAIL_ACCESS granted to GOATHUNTER
UAF>
```

Users not holding the identifier and trying to send mail through MX will see an error message stating that they are not authorized to send mail using MX.

5.1.1 System-wide MX Aliases for Outgoing Addresses

MX supports the use of personal and system-wide alias databases for defining aliases for frequently-used addresses for outgoing mail. The MXALIAS utility, described in the *Message Exchange User's Guide*, is used to maintain the database file.

To create a system-wide alias database, you can use the USE command in MXALIAS to open a centrally-located database file that can then be populated with the ADD command. To make the file accessible to users, set the protection to allow WORLD:READ access and define the MX_GLOBAL_ALIAS_DATABASE logical to point to the file:

```
$ set file/prot=w:re mx_dir:mx_global_alias.dat
$ define/system/exec mx_global_alias_database mx_dir:mx_global_alias.dat
```

You will also need to add the DEFINE command to your system startup.

5.1.2 VMS MAIL Protocol Prefix

MX by default uses the foreign protocol prefix MX% when interfacing with VMS Mail. You can define alternate foreign protocol prefixes for use with MX, to provide a migration path for users from other mail systems to MX. MX will correctly handle the following prefixes: SMTP%, WINS%, IN%, IHMF%, VN%, ST%, INET%, and UUCP%. To set up one of these alternate prefixes in VMS Mail, define the logical name MAIL$PROTOCOL_prefix:

```
$ DEFINE/SYSTEM/EXEC MAIL$PROTOCOL_prefix MX_MAILSHR
```

where **prefix** is one of the above-mentioned prefixes, without the trailing percent sign.

Note that incoming mail from MX will always bear the MX% prefix. If you wish to use another prefix for incoming mail, you can define the logical name MX_PROTOCOL_PREFIX:

```
$ DEFINE/SYSTEM/EXEC MX_PROTOCOL_PREFIX prefix%
```

where **prefix** is one of the above-mentioned prefixes, with the trailing percent sign. The default prefix MX% is the recommended prefix.

5.1.3 Default Host Names

By default, MX uses the logical name MX_NODE_NAME as the host name to be appended when converting a username into a full network mail address. For sites that require more control over how addresses are formatted, additional logical names may be defined.

These logical names are not automatically defined by MX; you must define them yourself (e.g., in your system startup procedure). The must be defined in executive mode, but may reside in the process, job, group, or system logical name table. The logical names and their descriptions are in Table 5–1.

Table 5–1 Address Formatting Logical Names

Logical name	Description
MX_ENVELOPE_ FROM_HOST	Host name to be used when formatting envelope return addresses (equivalent to the RFC821 MAIL FROM).
MX_FROM_HOST	Host name to be used when formatting the From: header.
MX_TO_HOST	Host name to be used when on a To: or Cc: address.

You can also affect address formatting by installing a SITE address conversion callout. See the *Message Exchange Programmer's Guide* for more information on address conversion callouts.

5.1.4 Character Set Conversion

MX_MAILSHR performs character conversion of the sender's VMS MAIL personal name and the VMS MAIL Subject: line from the local, native character set to an Internet standard character set for mail messages. The message body is similarly converted (by the Router agent).

By default, MX assumes that the local character set is ISO Latin-1 (ISO-8859-1), a widely used international standard. If you use the DEC Multinational Character Set (DEC-MCS), and cannot configure your terminals to use ISO Latin-1 instead, MX can automatically translate between DEC-MCS and ISO Latin-1 for you. You can activate this translation support by defining the following logical name logical name:

```
$ DEFINE/SYSTEM/EXEC MX_LOCAL_CHARSET_DEC_MCS any-value
```

The value of the equivalence string is unimportant; the existence of this logical name will identify the local character set as DEC MCS. Note that there is not a 1-to-1 mapping of all of the 8-bit characters in these two character sets.

If you use any other character set on the local system, or you wish to use a different network character set, you must install a custom character conversion callout module. See *Message Exchange Programmer's Guide* for further information about this customization.

5.2 SMTP_SERVER

This section describes the logical names used to configured the SMTP server.

5.2.1 SMTP Server Threads

The SMTP server is a detached, multi-threaded process. You can specify how many threads the server should handle simultaneously by defining a logical name:

```
$ DEFINE/SYSTEM/EXEC MX_SMTP_SERVER_THREADS n
```

The value of *n* should range from 1 to 32. The default is 4. The SMTP server may require larger process quotas/limits if more than four threads are allowed.

You can set the number of threads used by the SMTP server permanently by configuring this setting through the MXCONFIG.COM procedure.

5.2.2 Limiting "Outside" Thread Usage

If you have configured local IP networks (see Section 8.4.1), the MX SMTP server will limit the number of threads it uses to service clients sending messages from ''outside'' networks. This feature helps prevent local POP client users from getting get locked out of sending messages when there is heavy message traffic coming in from outside your network.

By default, the SMTP server limits the number of ‘‘outside’’ clients it will service at one time to 75% of the total number of threads configured. You may override this default by defining the following logical name:

```
$ DEFINE/SYSTEM/EXEC MX_SMTP_SERVER_MAX_OUTSIDE n
```

where *n* represents the maximum number of threads that the server may use to service ‘‘outside’’ clients. This number should range from 1 to the maximum number of configured threads.

5.2.3 Setting the SMTP Listener Port

By default, the SMTP server listens for incoming TCP connections on the default SMTP port (25). You can change this by defining the following logical name:

```
$ DEFINE/SYSTEM/EXEC MX_SMTP_PORT port-number
```

This should only be necessary for unusual configurations, such as having multiple SMTP servers running simultaneously on one system.

5.3 DECNET_SMTP Network Object

You must create a DECnet object called DECSMTP for establishing SMTP-over-DECnet connections. To do this, either use your mailer account or create a dedicated server account for use with the DECnet object (a dedicated server account is recommended). Using the AUTHORIZE utility, set a password for the this account and set the account /NOPWDLIFETIME. Also be sure the account has network access enabled.

```
UAF> MODIFY account/PASSWORD=some-password/NOPWDLIFETIME/network
```

A DECnet object needs to be created to handle the incoming SMTP-over-DECnet connections and to map the DECSMTP object name to a DECnet object number. Choose an unused DECnet object number. To see what object numbers are currently in use, use the command:

```
$ MCR NCP SHOW KNOWN OBJECT
```

Assign the object name DECSMTP to an unused object number; the number used *must* be identical on all nodes on your network that use SMTP-over-DECnet (this example uses 254). In NCP, use these commands:

```
NCP> PURGE OBJECT DECSMTP ALL
NCP> DEFINE OBJECT DECSMTP NUMBER 254 PROXY NONE FILE -
_NCP>    MX_EXE:DNSMTP_SERVER.EXE USER server-acct PASSWORD some-password
NCP> SET OBJECT DECSMTP ALL
```

You do not need to specify the FILE, USER, or PASSWORD parameters if you do not intend to accept incoming SMTP connections over DECnet. Be sure to use both the DEFINE and SET commands of NCP, and be sure that the password in the DECnet database matches the password you set for the server account in AUTHORIZE.

Using Proxies

Instead of storing the username and password for the server account in the DECnet database, you could grant access using DECnet proxies. Proxies give you more control over who on the network has access to the object, and eliminate the need for storing the password to the server account in the DECnet object database.

To enable proxy access to the DECSMTP object, use the following commands in NCP:

```
NCP> PURGE OBJECT DECSMTP ALL
NCP> DEFINE OBJECT DECSMTP NUMBER 254 PROXY INCOMING FILE -
_NCP>    MX_EXE:DNSMTP_SERVER.EXE
NCP> SET OBJECT DECSMTP ALL
```

Then in AUTHORIZE, create proxy entries for the mailer accounts on the other systems on the network that will be sending you mail via SMTP-over-DECnet:

```
UAF> ADD/PROXY remote::mailer server-acct/DEFAULT
```

For **remote::mailer** substitute the DECnet node of the remote system and the username of the mailer account on that system. For **server-acct** substitute the name of the server account you set up for use with the DECnet-SMTP object.

5.4 Message Entry Agent Shutdowns

The two message entry mechanisms that do not get shut down with the rest of MCP are the VMS Mail interface and the DECNET_SMTP server (if you are using SMTP-over-DECnet). The VMS Mail interface can be deactivated by de-installing the MX_MAILSHR image:

```
$ INSTALL REMOVE MX_MAILSHR
```

The SMTP-over-DECnet server gets shut down automatically when you shut down DECnet, or can be manually removed by eliminating the DECSMTP object from the DECnet database:

```
$ MCR NCP CLEAR OBJECT DECSMTP ALL
```

6 Managing the Message Queue

This chapter describes the various commands needed to control how the message queue operates.

6.1 Establishing the Queue Size

The maximum number of queue entries that can be present in the MX message queue at any one time is determined by the size, in blocks, of the MX message queue file. Each entry in the queue requires one block, with 10 additional blocks used to store a bitmap of entries in use. This means, for example, that a queue file that is 510 blocks in size will allow 500 entries to be present in the queue. The upper ceiling on the maximum entries is 131,072.

Most sites that process several thousand mail messages a day can probably work well with a queue file of about 5,000 blocks. If you are not short on disk space, creating a 131,072-block file will eliminate the need to ever modify the queue file size.

6.2 Running the MX FLQ Manager

As entries in the message queue are processed, they are marked as being finished. By default, one of the MX Router processes will be responsible for purging out finished entries.

You have the option of running a separate MX FLQ Manager process, whose sole job is to purge the queue of finished entries and cancel or ready any in-progress entries leftover from system crashes, disconnected processes, etc. Running a separate FLQ manager frees the MX Router to route messages, instead of splitting its time between routing and maintaining the queue. This means that the MX Router has more time for routing messages and queue maintenance isn't delayed while the MX Router is routing.

While the MX FLQ Manager can be run on multiple nodes in a cluster, only one manager is ever actively maintaining the queue. Running the manager on multiple nodes can provide failover backup in case of a node crash, etc. If the MX FLQ Manager is shutdown and there are no managers running on another node, one of the MX Router processes will automatically start maintaining the queue.

Sites that do not process many messages per day will probably not benefit from running the MX FLQ Manager process.

6.3 Queue Cleanup Logicals

The Router process (or the MX FLQ Manager process) automatically handles cleanup of the message queue. The time between cleanup events can be controlled with logical names, as described in Table 6–1.

Table 6–1 FLQ Manager/Router queue-related logicals

Logical	Default value	Description
MX_FLQ_MGR_ WAKEUP_INTERVAL	2 min.	Amount of time FLQ Manager sleeps before checking for entries to purge
MX_ROUTER_WAKEUP_ INTERVAL	10 min.	Amount of time MX Router sleeps before checking for entries to purge
MX_FLQ_CHECK_WAIT	10 min.	Amount of time between checks for other queue-related events
MX_FLQ_PURGE_WAIT	15 min.	Amount of time a queue entry should remain in queue after it has been processed

To alter one of these values, use the DEFINE command to set the logical to a new time (using VMS delta-time format) and send a reset signal to the Router and/or FLQ Manager processes:

```
$ DEFINE/SYSTEM MX_FLQ_PURGE_WAIT "0 00:10:00"
$ MCP RESET ROUTER,FLQ
```

(If the Router runs on a different node in the cluster, you will have to define the logical name there.)

If you want this change to be permanent and survive a system reboot, you should add this logical name definition to your system startup procedure before MX is started.

6.4 Automatic Purging of Finished Queue Entries

Finished queue entries are left in the queue for 15 minutes, by default, before they are purged. It is not necessary to leave the entries in the queue once they have been marked ‘‘FINished.’’ If you prefer to not leave them around, you can enable automatic purging of FIN entries and their related files. Use the MXCONFIG.COM command procedure, option 1, to configure the message queue logical names.

Even when autopurging is enabled, it is still necessary for the MX FLQ Manager or MX Router process to occasionally scan the queue for CANCELed entries. However, a dedicated MX FLQ Manager process is not as beneficial as it is when autopurging is not enabled.

6.5 The MCP QUEUE Commands

MCP includes a suite of commands for queue management to be used by privileged users. These commands are documented in the MCP command dictionary.

6.5.1 Interpreting MCP QUEUE SHOW Output

When there are messages in the queue, MCP QUEUE SHOW displays the following information about each entry:

```
Entry Status EntSize Source  Agent  Entry Status EntSize
----- ------ ------- ------ ------- ----- ------ -------
 2980 INPROG     229 LOCAL  <usr01@myhost.mycompany.com>
                            SMTP     2981 READY      229
                                    (waiting until 15-NOV-1991 15:07:21.75)
10859 READY   65120 LOCAL  <FileServ-Mgr@myhost.mycompany.com>
      (Waiting until 15-NOV-1991 18:00:00.00)
```

The fields of the display contain the following information:

- The first **Entry** field is the queue entry number for the base message, which can range from 1 to 131,071.
- The first **Status** field describes the status of the base message and can be one of INPROG, READY, FINISH, or CANCLD.
  - INPROG stands for ‘‘in progress’’ and is used when the base entry is being processed by the Router, or when one of its related entries is ready or in progress.
  - READY is used when the base entry is ready for processing by the Router.
  - FINISH indicates that processing of the base entry has completed. Finished entries remain in the queue for a short time before being removed (see Table 6–1). They are not normally displayed; the /ALL qualifier on the MCP QUEUE SHOW command can be used to force the display of these entries.
  - CANCLD indicates that processing of the entry is terminated before completion, such as when CTRL/C is pressed during entry of a message in VMS MAIL. Cancelled entries also remain in the queue for a short time before removal, and are only displayed when MCP QUEUE SHOW/ALL is used.
- The **EntSize** field displays the total size of the message entry. The size is calculated as the total number of bytes in the body of the message multiplied by the number of intended recipients of the message. Headers are not counted when computing the size of the message.
- The **Source** field describes the origin of the base message. It can have the value LOCAL, SMTP, DNSMTP (for SMTP-over-DECnet), SITE, or MAIL. To the right of the source display is the address of the user who originated the message.

If a message is being processed by one of the MX delivery agents, the base queue entry will be immediately followed by indented entries that begin with the **Agent** field. The **Agent** field identifies the delivery agent that is working on the entry. Possible values are LOCAL, SMTP, SITE, HOLD*n*, and DNSMTP (for SMTP-over-DECnet).

The second **Entry**, **Status**, and **Size** fields provide information about the queue entry used by the delivery agent. This agent-specific entry refers back to the base entry for the message headers and text, and the base entry has pointers to the agent-specific entries related to it. When an agent-specific entry is finished, the reference to it in the base entry is removed; when no agent-specific entries are left, the base entry is marked FINISHED.

6.5.2 Forcing a Message to "Bounce"

Occasionally, messages that can never be delivered will remain queued for multiple delivery attempts. When this occurs, you can force the message to be ''bounced'' back to the sender by using the command QUEUE READY/FINAL. This causes the delivery agent to make one final attempt to deliver a message; when that fails, it will be returned to sender.

For example, to force a final delivery attempt of the SMTP entry shown in Section 6.5.1, you would enter:

```
MCP> QUEUE READY/FINAL 2981
```

6.5.3 Interpreting MCP QUEUE STATISTICS Output

The MCP command QUEUE STATISTICS displays the following entry statistics:

```
MCP> QUEUE STATISTICS
Total entries: 16/502  (3%)   Highest entry used: 24  (4%)
MCP>
```

The first number after ''Total entries:'' is the current number of entries in the queue. The second number is the maximum number of entries allowed by the queue file size. The percentage of entries used is also shown.

The ''Highest entry used:'' is the largest entry number ever used during the life of the queue file. The percentage of the queue in use at that time is also shown. This value can be used to determine whether or not the selected queue file size is sufficiently large. The MCP command QUEUE EXTEND can be used to increase the size of the queue file.

7 Configuring MX for Part-Time Internet Connections

This chapter describes how to configure MX to use *holding queues* for messages that get delivered over a TCP/IP network path that is not available all the time.

7.1 Environment for Part-Time Connections

MX supports a messaging environment consisting of:

- A *server* system, located at a main office or Internet Service Provider, that has a permanent TCP/IP connection; and
- one or more *client* systems, located at remote offices, which periodically connect to the server system to pick up and deliver mail.

MX can act as either a client or server system; when configured as as a server, MX can service up to 32 clients with part-time network links. The client systems *must* have static IP addresses assigned to their domain names, or there must be an automatic assignment in the DNS between each client's domain name and its dynamic IP address.

7.2 Configuring MX as a Part-Time Client

To configure MX as a part-time client, use MCP to configure the delivery path to the server:

```
MCP> DEFINE PATH * HOLDING_QUEUE=1/ROUTE=server-name
```

If you have already used MXCONFIG.COM to create your MX configuration database, you may need to use the MODIFY PATH command to configure this path change.

This path configuration causes MX to hold all messages that are not delivered through other, more specific, paths.

If your client system creates the network path to the server, you should execute the following command to start the holding queue agent when the path is established:

```
$ @SYS$STARTUP:MX_STARTUP HOLD1
```

The agent will then attempt to deliver all messages in the holding queue to the server. You may also use this command if the server system creates the dial-up connection, but this may not be necessary if the server system supports the SMTP ETRN extension (documented in Internet RFC 1985). The MX SMTP server supports ETRN and can automatically start the holding queue agent when requested to do so by the server.

7.3 Configuring MX as a Server for Part-Time Clients

The configuration for MX as a server also uses holding queues, but defines paths for the specific host(s) or domain(s) at the client site:

```
MCP> DEFINE PATH client.host.name HOLDING_QUEUE=n/ROUTE=client.host.name
MCP> DEFINE PATH client.domain HOLDING_QUEUE=n/ROUTE=client.host.name
```

The value of *n* can be 1 through 32, representing one of the 32 available holding queues. If the name defined in the DEFINE PATH command does not match the client's actual Internet host name, you *must* specify the /ROUTE qualifier to set the host name to which messages will be delivered. In addition, all /ROUTE values for a single holding queue *must* be identical.

To start the delivery agent for one of the holding queues, use the command:

```
$ @SYS$STARTUP:MX_STARTUP HOLDn
```

where *n* is the number 1 through 32, representing the number of the holding queue for a particular client system. If a client system is also running MX, and is configured to start its holding queue agent when the network path is established, the ETRN support in the MX SMTP server will be used to automatically start the holding queue agent for their system.

7.4 ETRN Support

When both the client and the server systems run MX or another mailer that supports the RFC 1985 ETRN extension to SMTP, only one of the two systems has to be set up to start its delivery agent when the dial-up connection is active; the SMTP ETRN command will cause the other system to start delivering messages in the opposite direction.

A parameter is sent on the ETRN command; this parameter can identify the host or domain for which messages should be delivered, or it can be a special ''queue name,'' a private identifier used by the client and server systems to identify the holding queue agent that should be started. By default, MX holding queue agents will send the following ETRN commands:

```
ETRN mx-host-name
ETRN #ip-host-name
```

Where *mx-host-name* is the value of the logical name MX_NODE_NAME, and *ip-host-name* is the local system's TCP/IP host name (either from the logical name MX_INET_HOST, or from your TCP/IP package's configured host name). In the second case, the pound sign (''#'') indicates that *ip-host-name* is a queue identifier.

If the system at the other end of the connection is running MX, and has PATH definitions with /ROUTE qualifiers that specify the same host name as the *ip-host-name* appearing in the ETRN command, then the remote system will automatically start the appropriate holding queue agent for that system. The holding queue agent will also be started if the *mx-host-name* appearing in the first ETRN command matches one of PATH definitions.

You can override the default ETRN behavior for a particular holding queue agent with the following logical names:

```
$ DEFINE/SYSTEM/EXEC MX_HOLDn_REQUEST_DELIVERY FALSE
```

This logical name prevents the holding queue agent from sending any ETRN commands at all; use this when the remote system does not support ETRN or if you need to control the holding queue agents manually at both ends of the connection.

```
$ DEFINE/SYSTEM/EXEC MX_HOLDn_HOST_NAME "name"[,...]
```

This logical names overrides the parameters sent on the ETRN command. Multiple *name*s may be specified; enclose each in quotation marks, and separate them with commas. The value of each *name* is passed to the remote system exactly as entered; you may use the ''@'' or ''#'' prefix character to specify a domain name or a queue identifier (see RFC 1985 for more information). Use this logical name when the remote system supports ETRN but is *not* running MX, or if the IP host name used on the local system does not match the /ROUTE host names used in the MX configuration on the remote system.

7.5 Example Configuration

This example shows the commands used to configure a client servicing the domain ''mxclient.com'', with host name ''mailer.mxclient.com'', and a server with the host name ''mailhub.mxserver.com''.

Client configuration:

```
MCP> DEFINE PATH "mxclient.com" LOCAL
MCP> DEFINE PATH "mailer.mxclient.com" LOCAL
MCP> DEFINE PATH * HOLDING_QUEUE=1/ROUTE="mailhub.mxserver.com"
```

Server configuration:

```
MCP> DEFINE PATH "mxserver.com" LOCAL
MCP> DEFINE PATH "mailhub.mxserver.com" LOCAL
MCP> DEFINE PATH "mxclient.com" HOLDING_QUEUE=1/ROUTE="mailer.mxclient.com"
MCP> DEFINE PATH "mailer.mxclient.com" HOLDING_QUEUE=1/ROUTE="mailer.mxclient.com"
MCP> DEFINE PATH * SMTP
```

8 Reducing or Eliminating “Junk” E-Mail

This chapter describes the facilities in MX for preventing unwanted ‘‘junk’’ e-mail messages (also called ‘‘spam’’) from being processed.

8.1 What is “junk” E-Mail?

The Internet has grown explosively since its humble beginnings as a tool for researchers wishing to exchange information. The designers of the early Internet electronic mail systems and standards kept SMTP simple (in fact, the S in SMTP stands for ‘‘Simple’’), operating under the assumption that Internet users were, for the most part, well-behaved.

Due to the explosive growth, there are now many more people using the Internet that are not well-behaved, exploiting the openness of SMTP and the low cost of generating immense e-mail distribution lists either to annoy or to send unsolicited advertisements to the huge number of people who currently use Internet e-mail.

This use of e-mail, while costing almost nothing to the sender, can place a heavy burden—in terms of network and system resources—on the sites that receive these unsolicited messages. System managers wishing to reduce the number of unwanted ‘‘junk’’ messages can configure MX to block them from being received locally as well as prevent their systems from being used as a ‘‘relay’’ point, sometimes employed by unscrupulous ‘‘spammers’’ that have had their own systems blocked from direct delivery by other sites.

You can get more information about junk e-mail on the World Wide Web from *http://spam.abuse.net/spam*.

8.2 MX Filtering Features

MX contains several features to help curb junk e-mail:

- The MCP command SET SMTP/VALIDATE_SENDER_DOMAIN, which causes the SMTP server to validate the sender’s domain name on incoming messages;
- The MCP command SET SMTP/NORELAY, which rejects mail that does not originate at your site and is not destined for your site;
- The REJMAN utility, which manages a database of rejection criteria that you can populate with address and header patterns that have been used by spammers.
- Heuristic junk mail filters, also managed through the REJMAN utility, which filter potential junk mail based on the characteristics and/or contents of message headers in various combinations.

8.3 Sender Validation

Spammers often hide their identity by using a false return address on their messages. In many cases, the return address contains a domain name that does not actually exist in the Domain Name System (DNS). You can have the SMTP server validate the domain name appearing in SMTP MAIL FROM commands (which specify the return address for SMTP messages) to ensure that it exists in the DNS with the following MCP command:

```
MCP> SET SMTP/VALIDATE_SENDER_DOMAIN
```

This feature should only be enabled if you use the Internet and the Domain Name System. You should also ensure that all of your legitimate users that might send mail via SMTP to your system (such as PC users with POP mail clients) have their systems properly configured to send valid domain names as part of their return addresses.

Reliable DNS service is also helpful when enabling this feature; if the SMTP server cannot resolve a domain name due to communication problems with the DNS server, the validity check is *not* performed.

8.4 Disabling the SMTP Relay Function

In some cases, junk e-mail senders who are unsuccessful in getting messages delivered directly from their own systems, or who wish to obscure the source of the junk messages, take advantage of the ‘‘relay’’ function available in most SMTP servers. By default, most SMTP servers accept messages coming from any source and will attempt to deliver those messages to any destination, even if both the source of the message and its intended recipients are remote. This is called *relaying*. The MX SMTP server allows relaying by default; you can disable the relay function if you wish to prevent your system from being used as a relay point for junk e-mail:

```
MCP> SET SMTP/NORELAY_ALLOWED
```

Once relaying is disabled, the SMTP server checks the envelope FROM address (also called the Return-Path) of each incoming message to see if it is local or remote. It also checks each recipient address. If the envelope FROM address is remote and a recipient address is remote, delivery to the remote recipient is refused by the server.

The SMTP server uses the following tests to determine whether or not an envelope address (either FROM or recipient) is local or remote:

1 The host name in the address is compared against the MX host name. If both names are the same, or if the address being checked is the envelope FROM address and both hosts share the same parent domain, the address is considered local.

2 If the previous check did not classify the address as local, it is repeated using the TCP/IP host name.

3 The virtual domain address rewriter is called, if present, to see if the host name is on the virtual domain list.

4 If none of the above checks passes, the address is passed through the message routing rules (rewrite rules and path checks). If the resulting path is LOCAL, the address is considered local.

5 If none of the above checks classified the address as local, the host name is compared against the *local domains* list, which you configure with the DEFINE LOCAL_DOMAIN command.

You may need to use the DEFINE LOCAL_DOMAIN command if your host acts as a mail gateway for other systems. For example, if your system is part of the domain *mycompany.com* but also acts as a gateway for the domain *theircompany.com*, you would define the following local domains:

```
MCP> DEFINE LOCAL_DOMAIN theircompany.com
MCP> DEFINE LOCAL_DOMAIN *.theircompany.com
```

These commands will cause the SMTP server to treat any address of the form *user*@**theircompany.com** or *user*@*somehost.***theircompany.com** as being ‘‘local’’ for the purposes of relay-rejection testing.

8.4.1 Identifying Inside Networks and Hosts

You can further strengthen the anti-relay checks by using the DEFINE INSIDE_NETWORK_ADDRESS command to create a list of IP network and/or host addresses that are considered to be ‘‘inside’’ your local network. Creating this list prevents an outside system from using your system as a junk mail relay by masquerading as a local user (by using one of your local domain names in the SMTP MAIL FROM address).

If your system is acting as a relay between other hosts on your network(s) and the Internet, or you have POP or IMAP mail clients that send their messages through your system, you should add your network(s) to the list. For example, if your local network is 10.10.10.0, set up with a 24-bit subnetwork mask, you would specify

```
MCP> DEFINE INSIDE_NETWORK_ADDRESS 10.10.10.0/NETMASK=255.255.255.0
```

Any SMTP connection coming from a system on the 10.10.10.0 network would be allowed to specify a local domain in the SMTP MAIL FROM command in combination with a list of recipients that are outside your domain; all other systems would be considered ‘‘outside’’, and would be unable to specify a list of outside recipients even when sepcifying a local domain in the SMTP MAIL FROM.

If there were another mailer for your network (such as another system in the Domain Name System list of mail exchanger [MX] records for your domain) at the address 10.10.20.37, you would also specify

```
MCP> DEFINE INSIDE_NETWORK_ADDRESS 10.10.20.37
```

By omitting the /NETMASK qualifier, the address is assumed to be a host rather than a network.

If your system is the only mailer for your domain, and you do not have any POP or IMAP mail clients, you should specify

```
MCP> DEFINE INSIDE_NETWORK_ADDRESS 127.0.0.1
```

to enable the inside-address checks and prevent any other system from using a local e-mail address in the SMTP MAIL FROM in order to relay to outside hosts.

The DEFINE INSIDE_NETWORK_ADDRESS command has a /REJECT qualifier that allows you to set up specific exceptions to the list for individual hosts or subnetworks. For example, if on your local 10.10.10.0 network, there is one host (10.10.10.100) that should not be considered ‘‘inside’’, you could specify

```
MCP> DEFINE INSIDE_NETWORK_ADDRESS 10.10.10.100 /REJECT
```

to eliminate that host from the ‘‘inside’’ list.

Note that the list of addresses is kept in order based on the length of the network mask, from longest (host addresses) to shortest. The list is search sequentially from the beginning for a match against the sending host’s IP address.

8.4.2 Disabling the “Percent Hack”

To fully secure your SMTP server against relaying, you should also ensure that the MX Router has the percent-hack feature disabled:

```
MCP> SET ROUTER/NOPERCENT_HACK
```

This will protect your system from being used as a relay when presented with a local address that include a percent-sign in the username (a technique used for routing e-mail in the early days of the Internet).

To prevent your system from accepting *any* percent-hacked address, regardless of whether or not the domain-part of the address is local, use:

```
MCP> SET SMTP/NOPERCENT_HACK
```

With this setting, your system will refuse all percent-hacked addresses. This setting is necessary when your system accepts messages from outside sources and forwards them to other mailers that do not reject percent-hacked addresses.

Only combining all of the steps in this section can you ensure that your SMTP server is fully secured against relay attacks from junk mail senders.

8.5 Realtime Blackhole List

Several organizations maintain lists of known junk mail abusers and open relays, called Realtime Blackhole Lists (RBLs). These lists are typically made available through the Domain Name System so that SMTP servers can easily screen out messages coming from systems known to be sources or relay points for junk e-mail.

You can configure MX to perform an RBL domain name system check for every host connecting to the SMTP server. If the connecting host is found in the RBL, the SMTP server will refuse any message it tries to send.

Use the following MCP command to enable RBL checking:

```
MCP> SET SMTP/RBL_CHECK=(rbl-domain,...)
```

RBL checking is disabled by default. Consult your RBL provider for information on the DNS domain or domains it uses for its black-hole lists. You can configure the SMTP server to check multiple RBLs by specifying a comma-separated list of RBL domains.

8.6 Establishing Rejection Criteria with REJMAN

The REJMAN utility lets you create and manage a database containing criteria by which MX's SMTP server will reject incoming messages as junk e-mail that you do not want to be processed. See REJMAN for information on how to invoke this utility.

REJMAN provides commands for rejecting messages based on envelope contents (combinations of return address, recipient address, and the TCP/IP address of the sending host) as well as on headers contained in the message.

8.6.1 Address-Based Rejections

The REJMAN ADD REJECTION command can be used to configure the SMTP server to reject incoming messages with specific return addresses or matching specified patterns, or being sent from particular hosts. Messages matching the rejection criteria are refused by the SMTP server before the message contents are sent—reducing network load as well as the processing load on your system.

If your users are being bombarded with unwanted e-mail, review the unwanted messages for the Return-Path: header and the Received: headers (you should ensure that your SET LOCAL settings do not prevent those headers from being included in delivered messages). You can program a rejection rule into the SMTP server based on the Return-Path with:

```
REJMAN> ADD REJECTION sender-address
```

For example, if some junk e-mail had a Return-Path address of *<spammer@spamhost.com>*, you would use the command:

```
REJMAN> ADD REJECTION "spammer@spamhost.com"
```

If the Return-Path has different usernames, but the same host name, you could use a wildcard pattern:

```
REJMAN> ADD REJECTION "*@spamhost.com"
```

to reject all messages coming from *spamhost.com*.

If a large number of junk messages were all sent to your host from another single host or network, they would all have Received: headers that follow this pattern:

```
Received: from spam-source.com by yourhost.com with SMTP...
```

You can block incoming messages sent by a particular host or on hosts from a particular network using the /ADDRESS qualifier on the DEFINE REJECTION command. For example, if *spam-source.com* has an IP address of 10.0.0.1, you could use the command

```
REJMAN> ADD REJECTION "*@*"/ADDRESS=10.0.0.1
```

to have the SMTP server refuse all messages that were sent or relayed through that particular host. You must know the IP address of the host in order to use the /ADDRESS qualifier. Address specifications may also contain wildcards so you can block entire networks.

Once you have added one or more rejection rules to the database, you must use MCP to reset the SMTP server to have the new rules take effect:

```
MCP> RESET SMTP_SERVER
```

Note: Exercise caution when establishing rejection rules. An incorrect rule could block legitimate messages from reaching their intended destinations.

The ADD REJECTION command has more options for narrowing the scope of a rejection based on recipient addresses, for setting the text of the message returned when a rejection occurs, and for forwarding messages that would have been rejected to a different address to collect evidence of junk e-mail. It also includes support for using UNIX-style regular expressions for address matching, rather than VMS wildcards. See ADD REJECTION for more information.

8.6.2 Header-based Rejections

As spammers have become more sophisticated, junk e-mail messages have started looking more and more like legitimate messages, with perfectly valid envelope information. Such messages, however, often contain RFC822 message headers that can be used to identify them as junk. To block such messages, REJMAN provides for the addition of header-based rejection rules, with ADD REJECTION/HEADER:

```
REJMAN> ADD REJECTION/HEADER header-text-pattern
```

For example, some types of junk e-mail messages, advertising ‘‘get rich quick’’ schemes contain Subject: headers that begin and end with a string of dollar signs, as in:

```
Subject:  $$ MAKE MONEY FAST $$
```

Since it is unlikely that a legitimate message will contain a Subject: header that begins and ends with dollar signs, you might want to add the following header-based rejection to the REJMAN database:

```
REJMAN> ADD REJECTION/HEADER "Subject: $$*$$"
```

This will cause the SMTP server to reject any message containing a header that matches this pattern.

Note: As with address-based rejections, you should be careful when creating header-based rejection rules, to prevent the unwanted rejection of legitimate e-mail messages.

By default, header rejections match using VMS-style wildcards (‘‘*’’ and ‘‘%’’). Use a backslash to match one of these characters, rather than having it interpreted it as a wildcard. You can also use UNIX-style regular expressions instead of VMS wildcard pattern matching by using the /REGEX qualifier on the ADD REJECTION command, for more sophisticated matching rules.

For header-based rejections, the SMTP server receives the entire text of an incoming message before it can identify an unwanted message and return a status code to the sending system indicating that the message was not accepted (due to the way the SMTP protocol operates). Address-based rejections occur before any message data is sent.

Rejections based on header information can be redirected to other addresses (with the /FORWARD_TO qualifier). Unlike address-based rejection rules, however, you cannot modify the message returned to the sending system when a header-based rejection occurs.

8.6.3 Tracking Rejections

Tracking information is stored in the REJMAN database for each rejection rule. This information includes:

- the date and time you added a rule to the database;
- the date and time the SMTP server last rejected a message based on the rule; and
- a count of the number of times the rule has been used to reject a message

This information is mainly for your use, so you can gauge the effectiveness of the rejection rules you have entered. The SMTP server periodically updates the statistics in the database, and updates the database when it is reset or shut down.

8.6.4 Purging Old Rules

Since particular patterns of junk e-mail envelope and header characteristics tend to last only for a short time, REJMAN includes a PURGE command to allow you to delete rules from the database that have not been used recently:

```
REJMAN> PURGE [/BEFORE=date-time]
```

Unless you specify the /BEFORE qualifier, the PURGE command will remove rules from the database that have not been used to reject messages for 30 days or more.

Periodically using the PURGE command helps keep the database from growing too large, reducing the overhead involved in performing junk e-mail checks.

8.7 Heuristic Filters

Senders of junk e-mail use various techniques to disguise the source of their messages, and change their source addresses frequently. This makes it difficult to keep rejection rules up-to-date. In spite of these changes in addresses, most junk e-mail messages exhibit certain characteristics in the contents of their headers. The SMTP server contains a number of heuristic filters that look for these characteristics and classify matching messages as junk mail.

The filters are called ‘‘heuristic’’ because they determine only the *probability* that a message may be junk e-mail, based on rules created from empirical observation of thousands of junk e-mail messages. It is possible for legitimate, non-junk e-mail to match one or more of these filters, too. Because of this, the heuristic filters come with configuration options for forwarding matching messages to a mailbox (typically the local Postmaster or system manager) for further review, and for creating exceptions to the filters based on the sender’s address.

The REJMAN utility is used to configure the heuristic filters, using the commands described on the following pages:

- ENABLE HEURISTIC, ENABLE HEURISTIC
- DISABLE HEURISTIC, DISABLE HEURISTIC
- SET HEURISTICS, SET HEURISTICS
- ADD EXCLUSION, ADD EXCLUSION
- DELETE EXCLUSION, DELETE EXCLUSION

The REJMAN SHOW HEURISTICS command displays the current settings for the heuristic filters.

8.7.1 Confidence Levels

The heuristic filters provided by MX are listed in Table 8–1. Each filter is assigned a *confidence level* (CL) that represents, on a scale from zero to ten, the probability that a message matching the filter is junk mail. The default CL for each filter is shown in the table; you can change these values when configuring the heuristic filters.

When a message is received by the SMTP server, its headers are checked against each heuristic filter, and the message is assigned the highest CL found for all matching filters. When no filters match, the assigned CL is zero.

The message’s assigned CL is the checked against two configurable threshold values, the *accept* threshold and the *reject* threshold.

- If the assigned CL is *less than or equal to* the **accept** threshold, the message is classified as legitimate mail, and no further action is taken.
- If the assigned CL is *greater than* the **reject** threshold, the message is classified as junk mail, and is rejected.

- If the assigned CL is *between* the two thresholds, it is classified as ‘‘probable’’ junk; it is not rejected, but additional headers are inserted into the message to warn the recipient about its classification.

Table 8–1 Heuristic Filters

Filter name	Default CL	Description
FROM_TO_SENDER_SAME	10	Matches when the SMTP MAIL FROM: address matches both the From: and To: addresses in the RFC822 headers.
INVALID_AOL_ADDRESS	10	Matches when the RFC822 From: or To: headers contain an invalid address for AOL.COM (username too long or containing invalid characers).
INVALID_FROM	2	Matches when the From: header cannot be parsed.
INVALID_HOTMAIL_ADDRESS	10	Matches when the RFC822 From: or To: headers contain an invalid address for HOTMAIL.COM.
INVALID_TO	2	Matches when the To: header cannot be parsed.
MSGID_HAS_FROM	10	Matches when the Message-ID: header contains the RFC822 From: address.
MSGID_HAS_TO	10	Matches when the Message-ID: header contains the RFC822 To: address.
NULL_FROM	8	Matches when there is no RFC822 From: header in the message, or the header is present but empty.
NULL_MSGID	10	Matches when the RFC822 Message-ID: header contains a null message ID.
NULL_TO	6	Matches when there is no RFC822 To: header in the message, or the header is present but empty.
NUMERIC_ADDRESS	7	Matches when the username part of the From: or To: address contains only digits. Exceptions are provided for domains known to use all-numeric usernames, such as MCIMAIL.COM.

Table 8–1 (Cont.) Heuristic Filters

Filter name	Default CL	Description
PRECEDENCE_BULK	4	Matches when the message contains a Precedence: header containing the word “bulk”.
RECEIVED_AFTER_FROM	4	Matches when a Received: header is found after the From: header. This is usually an indication that the sender did not include a From: header in the original message, or that the sender forged the misplaced header.
RECEIVED_ALL_ZEROS	10	Matches when a Received: header is found after the From: header, *and* that Received: header contains the string “000.000.000.000”.
UIDL_AUTH_SENDER	10	Matches when the message contains an X-UIDL: header *and* a Comments: header that contains the string “authenticated sender is”.
X_UIDL	8	Matches when the message contains an X-UIDL: header. This header is normally used internally by some POP servers and clients to uniquely identify messages in mailboxes, and should not generally be found in outbound messages.

8.7.2 Rejection Actions

When a message’s confidence level exceeds the rejection threshold, the SMTP server applies the *rejection action* that you configure. This action can either be REJECT, which causes the SMTP server to refuse the message, or FORWARD, which causes the SMTP server to accept the message, but forward it either to the local Postmaster or to an address you specify.

By setting the rejection action to FORWARD, you can review messages that have been rejected by the filters and recover those messages that may have been misclassified. Each forwarded message will contain additional headers listing the original sender and recipients, plus a header that indicates the filter that caused the message to be rejected. If a legitimate message was misclassified, you can use the REJMAN ADD EXCLUSION

command to prevent future messages from the same originating address from being rejected.

8.7.3 Junk Mail Warnings

For a message that is identified as possible junk mail but is not rejected, the SMTP server inserts one or two headers to warn the recipient(s) that the message was identified as junk mail. The first header is:

```
X-Junk-Mail-Rating: {LOW, MEDIUM, or HIGH}
```

LOW indicates a CL less than 4; MEDIUM, 4 through 7; HIGH, 8 or higher. If your users have e-mail client programs that can also perform filtering, the addition of this header should make it simple for them to set their own policy regarding junk mail that is sent to them.

You may configure the SMTP server to include an additional header that identifies the filter that caused the message to be classified as junk mail. The REJMAN command that turns on this additional header is SET HEURISTICS/INCLUDE_REASON.

8.8 Logging SMTP Server Rejections

You can have the SMTP server notify you when it rejects a message due to sender validation, REJMAN rules, or heuristic checks by defining the logical name MX_SMTP_REJECTION_EVENT_CLASS:

```
$ DEFINE/SYSTEM/EXEC MX_SMTP_REJECTION_EVENT_CLASS opcom-class-name
```

If that logical name is defined as one of the OPCOM event class names, the SMTP server will log a notification each time it rejects an incoming message based on DNS validation of the sender address, or based on the rejection rules you have added to the REJMAN database.

8.9 Debugging Rejection Rules and Heuristic Filters

The SMTP server contains some additional logging code for debugging REJMAN rejection rules and heuristic filters. You can enable this debug logging with the following logical name definition:

```
$ DEFINE/SYSTEM/EXEC MX_ANTI_SPAM_DEBUG level
```

where *level* is an integer value greater than zero that specifies how much information should be included in the debug log. Basic information is included when the debug level is 1; more detail is included when the debug level is 2 or higher.

After defining this logical name, you must use MCP to RESET the SMTP server. To turn off debug logging, you should DEASSIGN the logical name and RESET the SMTP server. That will close the log file so you can examine the debug output.

8.10 Reducing Junk Mail Postings on Mailing Lists

Mailing lists can be configured to ignore postings that are suspected to be junk mail, by using the /IGNORE qualifier on the DEFINE LIST and MODIFY LIST commands. This qualifier takes two keywords, JUNK_MAIL, and MISSING_LIST_ADDRESS.

When a list posting matches one of the specified /IGNORE criteria, it is simply discarded, without being forwarded to the list subscribers or owners.

8.10.1 Ignoring Heuristically-Determined Junk Mail

When the heuristic junk mail detector in the SMTP server identifies, but does not reject, a message as possible junk mail, it adds the header ‘‘X-Junk-Mail-Rating:’’ to the message. This header contains a rating of LOW, MEDIUM, or HIGH which indicates the confidence level of the detection.

While you may wish to let such messages pass to individual users, and let them determine for themselves whether or not to discard the messages they receive, you may also want to be stricter about letting such messages become mailing list postings—especially for public mailing lists with a large number of subscribers. To cause the mailing list processor to ignore messages tagged by the heuristic junk mail detector, specify

```
/IGNORE=JUNK_MAIL[=level]
```

when creating or modifying the mailing list in MCP. If you do not specify a value for *level*, MEDIUM is assumed by default; this rejects all messages tagged with either a MEDIUM or HIGH confidence of being junk mail.

8.10.2 Ignoring Messages Without the List Address

When users post messages to a mailing list, they generally do so directly, so the address of the list appears in either the RFC822 ‘‘To:’’ or ‘‘CC:’’ header. Since many junk-mail generators do not include the intended recipient in either of those headers, you may wish to have the list processor ignore messages that do not directly include the mailing list’s address. To do this, specify /IGNORE=MISSING_LIST_ADDRESS when defining or modifying the list.

8.11 Using SMTP Authentication to Bypass Junk Mail Checking

Some sites have ‘‘roaming’’ users that need access to the mail system from outside the site’s network; such users have dial-up accounts with Internet Service Providers that assign addresses on an ‘‘outside’’ network. Mail sent by such users would normally be subject to all anti-relay and anti-junk-mail checking by the MX SMTP server, and would typically be rejected.

Roaming users that have e-mail client programs that implement the SMTP service extension for authentication (RFC2554) can authenticate to the MX SMTP server with a username and password, which will allow them to send messages that bypass all relay and junk mail checking.

8.11.1 Authentication Mechanisms Supported

MX supports authentication using the usernames and passwords in the VMS user authorization file through the PLAIN authentication mehchanism (RFC2222). MX also supports the non-standard LOGIN authentication mechanism, for compatibility with some clients that use it; it is similar to the PLAIN mechanism.

Use the following command to enable PLAIN and LOGIN authentication support:

```
MCP> SET SMTP/AUTHENTICATION=PLAIN
```

Note: The PLAIN and LOGIN authentication mechanisms do not encrypt the password that is provided by the client; they are sent to the server in the equivalent of plain text.

The MX SMTP server also implements the CRAM-MD5 authentication mechanism (RFC2195), using a private authentication database. To enable the CRAM-MD5 extension, use the following command:

```
MCP> SET SMTP/AUTHENTICATION=CRAM_MD5
```

With the CRAM-MD5 mechanism, passwords are one-way hashed before being sent to the server.

You can enable both extensions, if desired, by specifying a comma-separated list of keywords to the /AUTHENTICATION qualifier.

Note: The SMTP server only advertises its authentication support to clients that are not identified as being on the INSIDE_NETWORK_ADDRESS list.

Advertising authentication support causes some mail clients to require the user to enter a username and password, even if they are not needed to send messages. Since the authentication support only activates features that are needed by clients on outside networks, MX only advertises authentication support to those clients. This may change in a future release of MX.

8.11.2 Managing the SMTP Authentication Database

The hashing algorithm used by the CRAM-MD5 authentication mechanism is not compatible with the VMS user authorization system, so MX has its own database for storing the usernames and passwords used by the SMTP server. This database is kept in the file MX_DIR:MX_USERAUTH_DB.DAT by default. You may specify an alternate location for the database by defining a logical name:

```
$ DEFINE/SYSTEM/EXEC MX_USERUATH_DB file-spec
```

Use the MCP CREATE USER_DATABASE_FILE command to create an empty copy of the authentication database:

```
MCP> CREATE USER_DATABASE_FILE
```

This file is created with file protection that allows read and write access only to SYSTEM and OWNER.

Note: Do *not* change this default file protection! Only privileged users and programs should have any access to the authentication database!

Once the database file is created, you can use the ADD USER, MODIFY USER, and REMOVE USER commands to manage the usernames and passwords in the database. For example:

```
MCP> ADD USER "SmtpUser"/PASSWORD="SmtpUserPassword"
```

Usernames may be up to 16 characters long, and are case-sensitive. Only letters, digits, and underscores are allowed in usernames. Passwords may be up to 64 characters in length, and may contain any characters (although only printable characters are recommended); passwords are also case-sensitive.

8.11.3 How Authentication Works

When a compatible client connects to the SMTP server, it sends the SMTP EHLO (extended hello) command to find the extensions supported by the server. The SMTP server lists the AUTH extension in its reply; this informs the client that it may use the AUTH command to begin an authentication sequence.

When the client sends its authentication request using the AUTH command, the SMTP server uses the CRAM-MD5 mechanism to decode the authentication request and compares the result against the password against the one stored in the database. CRAM-MD5 uses a hash algorithm that prevents the username and password from being ‘‘sniffed’’ by a third party that might be monitoring the connection.

If the username and password are validated, any messages sent by the client are treated as though they came from a host on the INSIDE_ NETWORK_ADDRESS list, and will not be checked for relaying as long as the sender’s domain name is a local domain. In addition, no rejection rule or heuristic junk mail checks will be performed on the messages sent by the authenticated user.

Authentication failures are logged to the SMTP server log file, and are also logged to OPCOM if the MX_SMTP_AUTHFAIL_EVENT_CLASS logical name is defined.

8.11.4 PLAIN Authentication Features

By default, the PLAIN/LOGIN authentication mechanism uses the usernames and passwords in the VMS system authorization file (SYSUAF) for authenticating clients. Alternative authentication sources may be used by installing an authentication callout module. The *Message Exchange Programmer's Guide* has more information on writing and installing an SMTP authentication callout module.

8.11.5 Limiting SMTP Authentication Access to Certain Users

When using the default SYSUAF authentication mechanism for PLAIN/LOGIN, you may specify a VMS rights identifier that must be held by authenticating users for authentication to be completed, through a logical name defined in executive mode in the system logical name table:

```
$ DEFINE/SYSTEM/EXEC MX_SMTP_AUTH_IDENTIFIER identifier-name
```

When this logical name is defined, only those users who have been granted the specified identifier will be able to authenticate successfully to the SMTP server.

Note that this logical name is used only with the default SYSUAF mechanism for LOGIN/PLAIN SMTP authentication. It does not apply when an authentication callout is installed or when the private SMTP authentication database is used with the CRAM-MD5 authentication mechanism.

8.11.6 Intrusion Detection and Evasion

To help prevent password-guessing attacks on the system, the SMTP server implements intrusion detection and evasion measures. Some of the features described in the following sections apply to all authentication mechanisms; some only apply when the default SYSUAF mechanism is used with PLAIN/LOGIN authentication.

8.11.6.1 Authentication Retry Timers

When an authentication attempt fails, the authentication failure response to the client is delayed based on the LGI_RETRY_TMO system parameter, the same timer used for normal login password failures.

This timer is used for all authentication mechanisms.

8.11.6.2 Authentication Retry Limits

The SMTP server automatically disconnects the client when the number of authentication failures in a single SMTP session exceeds the count set by the LGI_RETRY_LIM system paramter, the same limit used for normal login retry limiting.

This limit applies for all authentication mechanisms.

8.11.6.3 Intrusion Evasion

When using the default SYSUAF mechanism for PLAIN/LOGIN authentication, the SMTP server tracks attempts to authenticate using invalid passwords, and implements intruder evasion measures similar to those used by VMS for normal logins. Table 8–2 lists the system parameters that the SMTP server uses for breakin tracking and evasion. Breakins are tracked by a combination of username and client IP address.

Table 8–2 Intrusion Detection and Evasion Parameters

System Parameter	Description
LGI_BRK_LIM	Limit on the number of failures for a single username/IP address combination before evasion measures begin. When evasion is started, even a valid password provided by the client is treated as an authentication failure.
LGI_BRK_TMO	Amount of time since the last authentication failure that must pass before a suspect is cleared from the intrusion list.
LGI_HID_TIM	Amount of time that evasive action persists after an intruder is detected.

These parameters are used only for the default SYSUAF mechanism for PLAIN/LOGIN authentication. They are not used if an authentication callout is installed (although the callout module may use them, if it is written to do so), nor are they used with the private CRAM-MD5 SMTP authentication database.

9 Other Miscellaneous Utilities

This chapter describes other utilities available with MX.

9.1 The MLFAKE Utility

For those times when you need to act on behalf of one of your users to sign off or subscribe to a mailing list, the MLFAKE utility may come in handy:

```
$ MLFAKE   :== $MX_EXE:MLFAKE
$ MLFAKE   listname   hostname   [command]  [arguments]
     /LISTSERV[=lsvname]
     /REQUEST=reqaddress
     /FROM=fromuser
```

Specify the name of the mailing list and its host (with no @ in between). If you omit **command**, it defaults to SIGNOFF. If the command requires additional arguments, you should specify them after **command** (in which case you must specify the command). If the mailing list is managed by an L-Soft LISTSERV, use the /LISTSERV qualifier; otherwise the request will go to the -Request address for the list (the Internet convention). You can override this altogether by specifying the request address with the /REQUEST qualifier. Finally, you must specify who the request is supposed to be from with the /FROM qualifier.

For example:

```
$ MLFAKE/FROM=someuser MX-List LISTS.WKU.EDU
$ MLFAKE/FROM=someuser/REQUEST="FileServ" -
_$          "" WKU.EDU SEND MX040.BLURB
```

The first example is for an Internet-type mailing list. The message will be constructed with ‘‘someuser’’ as the originator and MX-List-Request@lists.wku.edu as the destination, with the message reading SIGNOFF. The second example shows how MLFAKE can be used with file servers by specifying the destination user with the /REQUEST qualifier and omitting the **listname** argument (which is ignored when /REQUEST is specified).

MLFAKE requires SYSPRV privilege. SYSLCK privilege is not required, but will speed processing of the message. **DO NOT install the MLFAKE image with these privileges!** Only *trusted* users should have access to this utility, since it can be used to fake a mail message from any other user.

9.2 The MAILQUEUE Utility

MAILQUEUE is a program that scans the message queue for entries still in progress. It can be used by non-privileged users to view only those entries which were sent by them. When used from an account with SYSPRV privilege turned on, it lists all pending queue entries.

MAILQUEUE resides in the MX_EXE: directory and is designed to be executed as a DCL foreign command:

```
$ MAILQ*UEUE :== $MX_EXE:MAILQUEUE
$ MAILQ
```

If there are no delayed messages, MAILQUEUE returns the message

```
%MAILQ-I-MQNONE, no MX mail messages queued on local system
```

Otherwise, the MAILQUEUE display will resemble the following:

```
Entry: 9872, Origin: [SMTP] <SOMEUSER@yoyodyne.com>
  Status: IN-PROGRESS
  Local entry #9874, status: READY
      Waiting for retry until: 15-NOV-1991 16:46:44.12
      Recipient #1: SOMEUSER, Route=myhost.mycompany.com
      Error count=93
      Last error: %MAIL-E-OPENOUT, error opening !AS as output

Entry: 10859, Origin: [Local] <FileServ@myhost.mycompany.com>
  Status: READY, waiting until 15-NOV-1991 18:00:00.00
    Recipient #1: <SOMEUSER@somecompany.com>
```

9.3 The MX_DECODE Utility

The MX_DECODE utility will decode MIME-compliant mail messages with contents encoded as BASE64 or QUOTED-PRINTABLE. If the content type of the message is APPLICATION/VMS-RMS, it will also automatically restore the file's original RMS attributes. The MX Local agent automatically decodes VMS-RMS and QUOTED-PRINTABLE when they are received. MX_DECODE is provided for use in decoding messages delivered by other mailers, as well as for use with the MX Site agent, so that messages destined for MX Site may sent using SEND/FOREIGN.

MX_DECODE should be executed using a foreign command:

```
$ MX_DECODE :== $MX_EXE:MX_DECODE.EXE
$ MX_DECODE MAIL_MESSAGE.BASE64 XYZ.xxx
```

It accepts two required parameters: the input file and the output file. By default, in order to decode the file properly, the input file must include the MIME RFC822 headers before the encoded body. The headers are used only to find the stored VMS file attributes. If the file has a content type of APPLICATION/VMS-RMS, the resulting decoded output file will retain all of the VMS file attributes of the original file. Otherwise, the format of the resulting file will either be fixed-length, 512-byte records (for binary BASE64-encoded messages) or a text file (for QUOTED-PRINTABLE and BASE64-encoded text messages).

You can also decode base64-encoded files without the mail headers by using the qualifier /NOHEADER to tell MX_DECODE that there are no mail headers in the file, only encoded text. By default, MX_DECODE will produce a standard VMS binary file (fixed-length 512-byte records). If the text you're decoding is a text file, you can specify /TEXT to create a normal VMS text file. If it is a text file encoded using QUOTED-PRINTABLE, use the /QUOTED_PRINTABLE qualifier.

10 Troubleshooting MX

This chapter contains information on MX useful for debugging MX components.

10.1 Queue Files Used by MX Components

As has already been discussed, each MX component uses files in the message queue when processing messages. Each queue entry has at least one file associated with it, usually containing envelope information. The files created by MX are stored in a directory tree under the MX_FLQ_DIR: directory. The files are named *n.type*, where *n* is the queue entry number and *type* is a file type indicating the type of information is in the file.

There are ten subdirectories under the MX queue directory. The subdirectories are used to keep the size of the MX queue .DIR files below 128 blocks so that they can be cached by RMS. The subdirectory in which a file is located is determined by using the last digit in the file name as the subdirectory name ([.0], [.1], ..., [.9]).

Most of the queued files used by MX contain records written in tag-length-value (TLV) format. The tag and length fields are written in binary format, although the value may contain plain ASCII. While more efficient for MX, this storage format makes it more difficult to display the contents of these files, since the binary headers tend to confuse terminals. When examining these files, it is usually best to use DUMP or a text editor, rather than using TYPE.

10.1.1 File Types

The following list describes the file types used for queue files, the agents that write them, and the agents that read them.

SRC_INFO. This is the envelope information written on message entry. This file contains TLV records indicating the source of the message, the originating address, and the recipient addresses. Written by: MX_MAILSHR, DNSMTP_SERVER, SMTP_SERVER, MX_SITE_IN. Read by: MX_ROUTER.

HDR_INFO. This file contains the message headers, in TLV format. The headers are only used during address conversion when gatewaying mail into UUCP or for making return-address determinations on local delivery of mail. Written on message entry by: MX_MAILSHR, DNSMTP_SERVER, SMTP_SERVER, MX_SITE_IN. Read by: MX_LOCAL, MX_SMTP, MX_SITE, MX_MLF, MX_DNSMTP.

MSG_TEXT. This file contains the text of the body of the message, in plain ASCII. Written on message entry by: MX_MAILSHR, DNSMTP_SERVER, SMTP_SERVER, MX_SITE_IN. Read on message delivery by: MX_LOCAL, MX_SMTP, MX_SITE, MX_MLF, MX_DNSMTP.

DNSMTP_INFO, LOCAL_INFO, SMTP_INFO, SITE_INFO, MLF_INFO. These files contain envelope information used by the delivery agents. Written by: MX_ROUTER. Read by: MX_DNSMTP, MX_LOCAL, MX_SMTP, MX_SITE, MX_MLF (respectively).

Note that the SRC_INFO, HDR_INFO, and MSG_TEXT files remain attached to the original (base) queue entry. When the queue entries for the delivery agents are created, a back link to the original queue entry is entered so the delivery agents can gain access to the headers and message text. In addition, forward links to the delivery agent entries are kept in the original queue entry, which are zeroed out as each delivery agent finishes its processing. When all forward links are zeroed, the original queue entry is changed to FINISH status.

10.2 Process Names

The MX_START.COM command procedure assigns a specific process name to each of the MX detached processes. To determine whether an agent is running or not, use the MCP command STATUS or examine the SHOW SYSTEM output for the following process names:

MX Router	The Router
MX FLQ Manager	The MX queue manager
MX SMTP	SMTP delivery agent
MX DNSMTP	SMTP-over-DECnet delivery agent
SMTP Server	SMTP server
MX Local	Local delivery agent
MX MLF	Mailing list/file server
MX Site Agent	Site-specific interface agent
MX->SITE	Subprocess created by site interface

Note that the subprocesses are not created until at least one message is processed by the corresponding delivery agent.

10.3 Debug/Trace Output

Each of the delivery agents has debug/trace code that can be enabled to provide information on message processing. Tracing is enabled by defining a system-wide logical name, and disabled by deassigning that logical. Debugging can be enabled or disabled ‘‘on the fly’’: the process being debugged will automatically start logging trace information for each entry processed after the logical name is defined.

The trace log file, by default, is created in the same directory used for the agent’s main log file, with a file type of .LOG_*process-id* (for the SMTP server, the default file type is .LOG_*process-id_thread-id*). Trace output can be redirected by defining a system-wide logical name. The logical names used for debugging are outlined in Table 10–1.

There is no debugging code available in the MX_MAILSHR/MX_MAILSHRP (the VMS MAIL interface) or in MX_SITE_IN.

Table 10–1 Debug/Trace logical names

Agent	Enabling logical	Trace file	Default directory
Local	MX_LOCAL_DEBUG	MX_LOCAL_LOG	MX_LOCAL_DIR:
ML/FS	MX_MLF_DEBUG	MX_MLF_LOG	MX_MLF_DIR:
Router	MX_ROUTER_DEBUG	MX_ROUTER_LOG	MX_ROUTER_DIR:
Router/file queue	MX_FLQ_DEBUG	MX_FLQ_LOG	MX_ROUTER_DIR:
SMTP out	MX_SMTP_DEBUG	MX_SMTP_LOG	MX_SMTP_DIR:
SMTP server	MX_SMTP_SERVER_DEBUG	SMTP_SERVER_LOG	MX_SMTP_DIR:
SMTP-over-DECnet out	MX_DNSMTP_DEBUG	MX_DNSMTP_LOG	MX_DNSMTP_DIR:
SMTP-over-DECnet server	MX_DNSMTP_SERVER_DEBUG	DNSMTP_SERVER_LOG	MX_DNSMTP_DIR:
Site Agent	MX_SITE_DEBUG	MX_SITE_LOG	MX_SITE_DIR:

11 The MX Startup Process

This chapter describes the command procedures and files used by MX when it is started.

11.1 Startup Command Procedures

Typically, MX is started up by executing the command procedure SYS$STARTUP:MX_STARTUP.COM. This file is created at installation time simply to make MX easy to start; all it does is execute MX_ __ STARTUP.COM, which is located in the directory that eventually becomes the equivalence name for the logical name MX_EXE. MX_ __ STARTUP.COM contains the commands for setting up the MX logical names and invoking MX_START.COM, also located in the MX_EXE directory, to start the MX processing agents.

Individual MX components can be started by passing their names (one or more, separated with commas and with no intervening blanks) as arguments to SYS$STARTUP:MX_STARTUP.COM. Table 11–1 lists the components that the startup command procedures recognize.

Table 11–1 Component names for use with MX_STARTUP.COM

Name	Description
LOGICALS	Defines MX logical names and installs the MX shareable libraries.
NETLIB	Executes NETLIB's startup command procedure. (Prerequisite for ROUTER, SMTP, and SMTP_SERVER if using TCP/IP with MX.)
ROUTER	Starts the Router process.
LOCAL	Starts the local delivery agent.
SMTP	Starts the SMTP-over-TCP/IP delivery agent.
SMTP_SERVER	Starts the SMTP server (for TCP/IP).
DNSMTP	Starts the SMTP-over-DECnet delivery agent.
SITE	Starts the SITE interface.
MLF	Starts the mailing list/file server.

11.2 Startup Data Files

MX_ __STARTUP.COM uses two data files, both located in the MX root directory (MX_DIR:). MX_LOGICALS.DAT contains logical name definitions, some of which can be customized or altered after MX is installed. MX_STARTUP_INFO.DAT contains information on which of the MX components are installed, and on which nodes they should be run.

Do not edit these files directly. Use the MXCONFIG.COM procedure provided in the MX root directory to make any changes to the standard MX logical names and agent startup information.

MCP Command Dictionary

MCP

Executes the MX Control Program.

FORMAT

MCP *[command]*

Command Qualifiers	Defaults
/[NO]FILE=file-spec	*/FILE=MX_DIR:MX_CONFIG.MXCFG*

PARAMETERS

[command]

Any MCP command except the input redirection operator (@). The specified command is executed and control is returned to DCL immediately thereafter.

DESCRIPTION

MCP was written to be used as a DCL ''foreign'' command. To use it as a foreign command, you must define a symbol as follows:

```
$ MCP :== $MX_EXE:MCP
```

Defining the symbol in this way allows you to use the /FILE qualifier and specify ''one-shot'' commands on the command line.

By default, MCP loads in the current configuration file, MX_DIR:MX_CONFIG.MXCFG, when started.

QUALIFIERS

/[NO]FILE=file-spec

Loads the specified MX configuration file for editing. If not specified, MX_DIR:MX_CONFIG.MXCFG is loaded. The default file type is MXCFG. If /NOFILE is specified, MCP is started without loading any configuration information.

@ (Redirect Command Input)

Executes MCP commands read from a file.

FORMAT

@ *file-spec*

PARAMETERS

file-spec
Name of the file containing MCP commands. If omitted, the default file type is MCP.

DESCRIPTION

Use this command to have MCP take further command input from the specified file. There is no built-in limit on the number of levels of nesting of command files, so be careful when using input redirection from within a command file.

This command can only be used at the MCP command prompt, not as a ''one-shot'' MCP command. To have a file be used for input for an entire MCP session, use the following sequence of DCL commands.

```
$ DEFINE/USER SYS$INPUT file-spec
$ MCP
```

ADD USER

Adds a user to the SMTP authentication database.

FORMAT

ADD USER *username*

Command Qualifiers	Defaults
/PASSWORD=password-text	*/PASSWORD="PASSWORD"*

PARAMETERS

username

A username to be added to the database. Usernames may be up to 16 characters in length, and are case-sensitive. To add a lowercase or mixed-case username, surround it with quotation marks.

DESCRIPTION

This command adds a username to the authentication database used by the SMTP server. Any SMTP client that authenticates to the SMTP server with a valid username and password can send messages via the SMTP server as if they were connected to an ''inside'' network, even if they are currently not at an address on the INSIDE_NETWORK_ADDRESS list; all anti-junk-mail and anti-relay checking is disabled.

QUALIFIERS

/PASSWORD=password-text

Specifies a password for this username. Passwords can contain any characters, and are case-sensitive. They may be up to 64 characters in length. If you do not specify a password on the ADD USER command, a default password, ''PASSWORD'' (in upper case), is assigned.

ATTACH

Transfers control to another process in the current process tree.

FORMAT

ATTACH *[process-name]*

Command Qualifiers	Defaults
/IDENTIFICATION=process-id	
/PARENT	

PARAMETERS

process-name

Name of the process to which the terminal should be attached. The process must be in the current process tree. This parameter is omitted if one of the qualifiers is specified.

DESCRIPTION

This command is similar to the DCL ATTACH command. It is used in interactive jobs to attach the terminal to another process in the current process tree for the job.

QUALIFIERS

/IDENTIFICATION=process-id

Specifies the process by its process identification, a hexadecimal number.

/PARENT

Specifies that the terminal should be attached to the current subprocess's immediate parent in the process tree.

CREATE USER_DATABASE_FILE

Creates a new version of the SMTP authentication database.

FORMAT

CREATE USER_DATABASE_FILE *[file-spec]*

PARAMETERS

file-spec

Name of the file to be created. If omitted, a new version of the main authentication database file, MX_DIR:MX_USERAUTH_DB.DAT, is created.

DESCRIPTION

This command should typically be used only once, to create the SMTP authentication database for use by the SMTP server and the MCP ADD USER, MODIFY USER, and REMOVE USER commands.

DEFINE/KEY

Defines an equivalence string and a set of attributes with a key on the terminal keyboard.

FORMAT

DEFINE/KEY *key-name equivalence-string*

Command Qualifiers	Defaults
/ECHO	
/ERASE	
/IF_STATE	
/LOCK_STATE	
/LOG	
/SET_STATE	
/TERMINATE	

DESCRIPTION

See the DCL help entry for DEFINE/KEY for more information on this command.

You can have a set of keys defined automatically for use with MCP by placing DEFINE/KEY commands in the file SYS$LOGIN:MX_MCP_KEYDEFS.INI. Note that this is the same file that is used with the REJMAN program.

DEFINE ALIAS

Defines a local alias for transparent mail forwarding.

FORMAT

DEFINE ALIAS *local-name fwd-address[,...]*

PARAMETERS

local-name

A string up to 32 characters in length. Any E-mail addressed to this name on the local host will be sent to the forwarding address.

fwd-address

A valid E-mail address, which will be substituted for the matching local alias address. Multiple addresses may be given; use commas to separate the addresses. The maximum character length for all addresses is 255 characters, including commas.

DESCRIPTION

An alias can be used to cause mail messages to be forwarded automatically to another address. Unlike forwarding using the SET FORWARD command in VMS Mail, no ''Resent'' headers are added to the message. In addition, alias-based forwarding is performed by the MX routing agent rather than the local delivery agent, thus affording a small savings in message queue space and processing time. Due to the lack of notification, however, it is recommended that aliases be used sparingly.

DEFINE FILE_SERVER

Creates a file server.

FORMAT

DEFINE FILE_SERVER *name*

Command Qualifiers	Defaults
/BEGIN_SEND_PERIOD=hh:mm	
/[NO]DELAY_THRESHOLD=size	
/[NO]DESCRIPTION=text	*/NODESCRIPTION*
/END_SEND_PERIOD=hh:mm	
/[NO]HOST_LIMIT=hostlim	
/[NO]MAILING_LIST=listname	
/MANAGER=address	
/ROOT=rootspec	
/[NO]SERVER_LIMIT=servlim	
/[NO]USER_LIMIT=userlim	

PARAMETERS

name
Local name to be used for the file server, up to 32 characters in length.

DESCRIPTION

This command is used to establish or remove an MX mail-based file server on the local system. The server can be set up to distribute groups of files called ‘‘packages’’ using E-mail as the distribution medium. The file server responds to commands placed, one per line, in the text of a mail message sent to the file server username. The commands the file server responds to are HELP, LIST, SENDME, QUIT, and ADDRESS.

The root you specify with /ROOT qualifier is used by the file server software to locate packages. Each package must have a directory [package-name] under that root where all its files are kept. In addition, the file name of each of the files in the package must also match the package name. Each package must also have a file called package-name.DESCRIPTION in the top-level root directory that contains a description of the package and the files in the package.

The .DESCRIPTION files may be placed in the package subdirectories, if desired, but they cannot exist in both the root and the subdirectories.

The SENDME command takes one argument, the name of a package or an individual file. If a package name is specified, all files in the package directory are sent to the requesting user. Otherwise, just the specified file is sent.

The LIST command can take a wildcard pattern as an argument (if omitted, it defaults to ‘‘*’’). The contents of the description files of all packages whose names match the wildcard pattern are placed in a file and sent to the requesting user.

The HELP command causes the file server to send the file FILESERV_HELP.TXT from the top-level root directory to the requesting user. A sample help file is provided with MX, which the system manager can modify to provide site-specific information.

The QUIT command causes the file server to ignore any remaining lines in the message. It can be used to prevent the unintentional parsing of mail signatures.

The ADDRESS command takes a valid RFC822-compliant address. It causes all file server replies to be redirected to the given address instead of the Reply-To or From addresses.

QUALIFIERS

/BEGIN_SEND_PERIOD=hh:mm

Identifies the time of day when the file server can begin sending files that exceed the delay threshold size. Defaults to 17:00.

/[NO]DELAY_THRESHOLD=size

Use /DELAY_THRESHOLD to establish the maximum size, in bytes, a file can be to be sent at any time during the day. Files exceeding *size* are sent only during the sending period established by /BEGIN_SEND_PERIOD and /END_SEND_PERIOD. Use /NODELAY_THRESHOLD to remove size restrictions.

/[NO]DESCRIPTION=text

This qualifier defines a brief description for the file server. This description is added to the file server address in the X-FileServer header on outgoing server messages.

/END_SEND_PERIOD=hh:mm

Identifies the time of day when the file server should stop sending files that exceed the delay threshold size. Defaults to 09:00.

/[NO]HOST_LIMIT=hostlim

Specifies that a maximum of *hostlim* bytes may be sent per day to any single host.

/[NO]MAILING_LIST=list-name

Specifies a mailing list to be linked to the file server. Only those users who are subscribed to the specified list may have access to the file server. The specified list must exist on the local system in order for this qualifier to have any effect.

/MANAGER=address

When establishing a file server, you can provide an E-mail address to which all error messages and mail that bounces back to the file server can be forwarded. The local alias *name*-Mgr will be created to direct those error messages to the /MANAGER address. If you omit the /MANAGER qualifier, bounced mail will be directed to the Postmaster.

/ROOT=rootspec

You must specify a location (either a rooted logical or a device plus root directory specification) to be used as the root for the file server files and directories. Examples of valid roots are FILESERV_ROOT: (if it is defined

as a rooted logical) and DISK:[FILE_SERVER.] (note the final dot before the bracket, indicating it is a root specification).

/[NO]SERVER_LIMIT=servlim

Specifies that a maximum of *servlim* bytes may be sent per day from the server.

/[NO]USER_LIMIT=userlim

Specifies that a maximum of *userlim* bytes may be sent per day to any one user.

DEFINE INSIDE_NETWORK_ADDRESS

Defines an "inside" network address for SMTP relay determination.

FORMAT

DEFINE INSIDE_NETWORK_ADDRESS *ip-address*

Command Qualifiers	Defaults
/NETMASK=ip-netmask	*/NETMASK=255.255.255.255*
/[NO]REJECT	*/NOREJECT*
/[NO]RELAY_ALLOWED	*/NORELAY_ALLOWED*

PARAMETERS

ip-address

IP address of the network or host to be added to the list, in dotted-decimal form. The address is assumed to be for a host unless the /NETMASK qualifier is specified.

DESCRIPTION

This command establishes an IP address or network that is in one of the local domains, is permitted to use your SMTP server as a relay, or to reject a particular host or network from being considered as part of your local domain.

Inside network address definitions are only used with the SMTP server is set to disallow relays with SET SMTP/NORELAY_ALLOWED. When at least one inside address is defined, messages coming in via SMTP are allowed to have recipients outside of the local domain(s) only if the sending system's IP address is on the inside network address list.

By default, the SMTP server will still reject a message that contains non-local addresses for both the sender and the receiver, even from hosts on the inside network address list. You can ease that restriction with the /RELAY_ALLOWED qualifier.

QUALIFIERS

/NETMASK=ip-netmask

Specifies the network mask to be applied to the address, in dotted-decimal form. The default is 255.255.255.255, which indicates that the IP address is for a host, not a network.

/[NO]REJECT

Indicates whether relay is to be rejected from the specified host or network. This qualifier can be used to reject SMTP relay from particular hosts or subnetworks that are below a parent network that is already on the inside network address list.

/[NO]RELAY_ALLOWED

Indicates that the host(s) should be allowed full relay permission; that is, messages sent from the host(s) are allowed to contain non-local addresses for both sender and receiver.

This qualifier is useful when your system is acting as a central mail hub, and there are hosts on your local network that automatically forward messages for their local users to hosts outside your domain via an alias. When such messages are sent back to your system (as the mail hub), they will contain non-local addresses for both the sender and the recipient.

DEFINE LIST

Creates a mailing list.

FORMAT

DEFINE LIST *list-name*

Command Qualifiers	Defaults
/[NO]ADD_MESSAGE=fspec	*/NOADD_MESSAGE*
/[NO]ARCHIVE=fspec	*/NOARCHIVE*
/[NO]CASE_SENSITIVE	*/CASE_SENSITIVE*
/[NO]CC_POST_ERRORS	*/NOCC_POST_ERRORS*
/[NO]CONFIRMATION_MESSAGE=fspec	*/NOCONFIRMATION_MESSAGE*
/[NO]DESCRIPTION=text	*/NODESCRIPTION*
/[NO]DIGEST	*/NODIGEST*
/ERRORS_TO=address	*See text*
/[NO]FORWARD_MESSAGE=fspec	*/NOFORWARD_MESSAGE*
/[NO]HIDE_ERRORS_TO	*/HIDE_ERRORS_TO*
/[NO]HOSTNAME=hostname	*/NOHOSTNAME*
/IGNORE[=(keyword,[...])]	*See text*
/[NO]LIST_HEADERS=(keyword[,...])	*/NOLIST_HEADERS*
/[NO]MAXIMUM_MESSAGE_SIZE=limit	*/NOMAXIMUM_MESSAGE_SIZE*
/[NO]MODERATOR=(address[,...])	*/NOMODERATOR*
/[NO]NOTIFY=(keyword,[...])]	*/NONOTIFY*
/OWNER=(address[,...])	
/PRIVATE	*/NOPRIVATE*
/PROTECTION=prot-spec	*See text*
/[NO]RECIPIENT_MAXIMUM={DEFAULT | n}	*/RECIPIENT_MAXIMUM=DEFAULT*
/[NO]REMOVE_MESSAGE=fspec	*/NOREMOVE_MESSAGE*
/REPLY_TO=(kwd[,...])	*/REPLY_TO=SENDER*
/[NO]REQUEST_CONFIRMATION[=INTERVAL=delta-time]	*/NOREQUEST_CONFIRMATION*
/[NO]RETURN_ADDRESS=address	*See text*
/SETTINGS=(kwd[,...])	*/SETTINGS=DEFAULT*
/STRIP_HEADERS=keyword	*See text*
/SUBJECT_PREFIX=string	*/NOSUBJECT_PREFIX*
/[NO]TEXT_ONLY	*/NOTEXT_ONLY*
/XHEADERS=(string[,...])	*/NOXHEADERS*

PARAMETERS

list-name
Local name to be used for the mailing list, up to 32 characters in length.

DESCRIPTION

This command is used to establish a mailing list. When a message is sent to the mailing list address, the mailing list processor forwards a copy of the message to all the addresses on the list. In addition, it can place a copy of the message in a file, called an archive.

Mailing lists are fully described in *Message Exchange Mailing List/File Server Guide*.

QUALIFIERS

/[NO]ADD_MESSAGE=fspec

Specifies the name of a file to be sent to a user subscribing to the list. If omitted, the device and directory default to MX_MLIST_DIR: (MX_ROOT:[MLF.MAILING_LISTS]), and the file type defaults to TXT.

The default for this qualifier is /NOADD_MESSAGE, which causes the global add message, MX_MLIST_DIR:MLIST_ADD_MESSAGE.TXT, to be sent when a user subscribes to the list. See *Message Exchange Mailing List/File Server Guide* for more information about notification messages.

/[NO]ARCHIVE=fspec

Specify /ARCHIVE to have the mailing list messages placed in an archive file automatically by the mailing list processor. For *fspec* you must provide at least a device/directory specification. If the file name is omitted, the mailing list name is used as the file name for the archive file. If the file type is omitted, *yyyy-mm* is used as the file type, where *yyyy* is the current year and *mm* is the number of the current month at the time a message is added to the archive.

/[NO]CASE_SENSITIVE

This qualifier enables or disables case-sensitivity with regard to mailing list subscribers. By default, MX treats the left-hand side of subscriber addresses in a case-sensitive manner with regard to SIGNOFF and SET commands. If a list is defined /NOCASE_SENSITIVE, then the case of subscriber addresses will be ignored.

/[NO]CC_POST_ERRORS

This qualifier enables or disables copying mailing post failure messages to the /ERRORS_TO address. By default, if a message cannot be forwarded to a list, an error message is sent back to the sender of the message. If /CC_POST_ERRORS is set, those error messages are also sent to the /ERRORS_TO address. This lets the list owner see attempted posts from non-subscribers and other posting failures.

/[NO]CONFIRMATION_MESSAGE=fspec

Specifies the name of a file to be sent to a potential subscriber when the list is set for subscription confirmation requests. If omitted, the device and directory default to MX_MLIST_DIR: (MX_ROOT:[MLF.MAILING_LISTS]), and the file type defaults to TXT.

The default for this qualifier is /NOCONFIRMATION_MESSAGE, which causes the global confirmation message, MX_MLIST_DIR:MLIST_CONFIRM_MESSAGE.TXT, to be sent for any confirmation requests. See *Message Exchange Mailing List/File Server Guide* for more information about confirmation messages.

/[NO]DESCRIPTION=text

This qualifier defines a brief description for the mailing list. This description is added to the mailing list address in the X-ListName header on list messages.

/[NO]DIGEST

This qualifier enable or disables digest support for the list. A list marked /DIGEST can support the DIGEST flag for subscribers. Mail sent to the "-digest" form of the list address will be forwarded only to those subscribers marked as digest subscribers.

/ERRORS_TO=address

This qualifier is used to direct error messages and mail returned to the mailing list processor to the specified address. If not specified, the address of the the first specified owner of the mailing list is used.

/[NO]FORWARD_MESSAGE=fspec

Specifies the name of a file to be sent to a user subscribing to the list when the list does not have W:E access set. The message should notify the user that the subscription request was forwarded to the list owner. If omitted, the device and directory default to MX_MLIST_DIR: (MX_ROOT:[MLF.MAILING_LISTS]), and the file type defaults to TXT.

The default for this qualifier is /NOFORWARD_MESSAGE, which causes the global forward-to-owner message, MX_MLIST_DIR:MLIST_FORWARD_MESSAGE.TXT, to be sent when a user tries to subscribe. See *Message Exchange Mailing List/File Server Guide* for more information about notification messages.

/[NO]HIDE_ERRORS_TO

Controls how the mailing list processor formats the envelope FROM address and Errors-To: header. When a list is set /HIDE_ERRORS_TO (the default), the address specified with the /ERRORS_TO qualifier (or the first /OWNER address, if /ERRORS_TO is not specified) is replaced automatically in outbound list postings by the alias ''owner-*listname*''. Setting /NOHIDE_ERRORS_TO prevents this substitution, using the errors-to address explicitly.

/[NO]HOSTNAME=hostname

The /HOSTNAME qualifier specifies the host name to be used for the mailing list's address. By default, the local MX host name is used. An alternative host name can be used when the MX installation is being used to provide mail service for multiple domains. You must configure MX to route the host name via the LOCAL path.

/IGNORE[=(keyword,...)]

The /IGNORE qualifier instructs MLF to ignore postings to the mailing list if they match the specified criteria. The criteria keywords are MISSING_LIST_ADDRESS and JUNK_MAIL. These keywords are negatable. By default, no postings are ignored.

Specifying the MISSING_LIST_ADDRESS criterion causes MLF to ignore postings to the list that do not explicitly include the list's address in either the To: or CC: header of the message. This keyword does not take a value. MLF performs address rewriting and alias translation on the header addresses, so a locally-defined rewrite rule or alias that results in the mailing list address is considered valid.

Specifying the JUNK_MAIL criterion causes MLF to ignore postings that contain the X-Junk-Mail-Rating: header that is inserted by the heuristic junk-mail filter in the SMTP server. This keyword takes a value: LOW, MEDIUM, or HIGH, corresponding to the confidence level of the likelihood that the message is junk mail, as entered in the X-Junk-Mail-Rating: header by the SMTP server. Only those messages with the specified or higher rating are ignored; i.e., if MEDIUM is specified as the keyword value, only those messages with MEDIUM or HIGH ratings are ignored.

/[NO]MAXIMUM_MESSAGE_SIZE=limit

The /MAXIMUM_MESSAGE_SIZE qualifier instructs MLF to reject any messages posted to the list whose message contents (excluding headers) exceeds *limit* Kbytes. Disabled by default. Specifying zero for *limit* has the same effect as specifying /NOMAXIMUM_MESSAGE_SIZE.

/[NO]LIST_HEADERS=(keyword[=value][,...])

The /LIST_HEADERS qualifier instructs MLF to include or omit special X-List-* headers that provide URLs for subscribing to a list, unsubscribing from a list, and getting help for that list.

There are three valid keywords: SUBSCRIBE, UNSUBSCRIBE, and HELP. All three accept values that are used in the creation of the actual headers, which will be added to each message posted to the mailing list. However, only HELP requires a value. If the value is omitted for SUBSCRIBE and UNSUBSCRIBE, the proper URLs for those actions will be automatically generated by MLF.

The List-* headers are currently a proposed Internet standard. The actual headers generated are X-List-Subscribe, X-List-Unsubscribe, and X-List-Help.

Clients that support these headers (both X-List-* and List-*) will provide click buttons to perform the specified actions (usually "mailto" URLs).

See *Message Exchange Mailing List/File Server Guide* for more information on mailing list headers.

/[NO]MODERATOR=(address[,...])

This qualifier is for future use. Moderated mailing lists are currently not supported.

/[NO]NOTIFY=(keyword[,...])

This qualifier specifies owner notification options. Valid keywords are ALL, ADD, REMOVE, REQUEST, and SET; descriptions are given in Table MCP–1. If you specify ALL, no other keywords are allowed. By default, no owner notifications are sent.

Table MCP–1 Owner Notification Types

Keyword	Description
ALL	Notifications sent for all events listed below.
ADD	Notification sent to owner(s) when an address is successfully added to the list.
REMOVE	Notification sent to owner(s) when an address is successfully removed from the list.
REQUEST	Notification sent to owner(s) when a subscription confirmation request is sent or expired.
SET	Notification sent to owner(s) when a subscriber's settings have been changed with a SET or MODIFY mailing list request.

/OWNER=(address[,...])

This qualifier specifies the addresses of one or more owners of the mailing list. Each mailing list must have at least one owner, who is responsible for

handling subscription requests not handled automatically by the mailing list processor and problems with or questions about the list.

/[NO]PRIVATE

This qualifier specifies that the list is private and should not be displayed in response to DIRECTORY commands sent to the MXserver or -Request addresses. The list protection is not affected by this qualifier.

/PROTECTION=prot-spec

This qualifier determines the protection of the mailing list. The protection specification, *prot-spec*, is identical to a VMS file protection specification, and defaults to (S:RWED,O:RWED,G:RWED,W:RWE). The four protection classes are described in Table MCP–2 and the four protection types are described in Table MCP–3.

Table MCP–2 Mailing list protection classes

Class	Description
SYSTEM	any address matching one of the addresses on the system user list (see DEFINE SYSTEM_USERS)
OWNER	any address matching one of the owner addresses specified on the /OWNER qualifier
GROUP	any address matching one the addresses on the subscriber list for the mailing list
WORLD	any other address

Just as with VMS file protections, the SYSTEM and OWNER classes are implicitly granted C (control) access, allowing them to use the ADD and REMOVE commands to add and remove addresses from the mailing list.

Table MCP–3 Mailing list protection codes

Code	Description
R (Read)	allows the use of the REVIEW command
W (Write)	allows the user to post messages to the list
E (Enroll)	allows the automatic handling of the SUBSCRIBE command
D (Delete)	allows the automatic handling of the SIGNOFF command

Note that protection code E (enroll) is only meaningful when used with the WORLD class and that protection code D (delete) is only meaningful when used with the GROUP class.

Some typical GROUP and WORLD protection specifications are shown in Table MCP–4. In most cases, you would also want to give SYSTEM and OWNER users RWED access.

Table MCP–4 Typical protection codes

(G:RWED,W:RWE)	Public list. Anyone can subscribe, sign off, and review the list; anyone can post to the list.

Table MCP–4 (Cont.) Typical protection codes

(G:RWED,W:E)	Semi-public list. Anyone can subscribe and sign off the list, but only subscribers can review or post to the list.
(G:W,W)	Private list. Only subscribers can post to the list, and all subscription requests are screened by the owners of the mailing list.
(G,W)	One-way list. Only the owners can post to the list, and they also screen all the subscription requests.

Note: Since electronic mail can readily be forged, you should not depend on this protection scheme for absolute security of your mailing lists. The mailing list processor attempts no authentication of addresses when it receives messages.

/[NO]RECIPIENT_MAXIMUM={DEFAULT|n}

Specifies the maximum number of recipients to be entered per outbound mailing list message for the list being defined. The default setting, /RECIPIENT_MAXIMUM=DEFAULT, causes this setting to be taken from the SET MLF/RECIPIENT_MAXIMUM setting. Specifying /NORECIPIENT_MAXIMUM causes a single outbound message to be created for each list posting, with all recipients listed. You may also specify a postive integer for *n*, which instructs MLF to enter no more than *n* recipients in a single message. If the mailing list has more than *n* subscribers, each list posting will cause multiple, duplicate messages to be generated, each with no more than *n* recipients.

/[NO]REMOVE_MESSAGE=fspec

Specifies the name of a file to be sent to a user signing off the list. If omitted, the device and directory default to MX_MLIST_DIR: (MX_ ROOT:[MLF.MAILING_LISTS]), and the file type defaults to TXT.

The default for this qualifier is /NOREMOVE_MESSAGE, which causes the global remove message, MX_MLIST_DIR:MLIST_REMOVE_ MESSAGE.TXT, to be sent when a user signs off the list. See *Message Exchange Mailing List/File Server Guide* for more information about notification messages.

/REPLY_TO=(kwd[,...])

Specifies how the mailing list processor should handle Reply-To headers. Available reply-to types are SENDER and LIST, which may be combined. The default is SENDER, which prevents the mailing list processor from modifying the headers. If LIST is specified, a Reply-To header is added to list messages to re-direct replies to the mailing list, eliminating any existing Reply-To header in the original message. If LIST and SENDER are both specified, a Reply-To header containing both the mailing list address and the original Reply-To address is added to list messages (using the From address if no Reply-To header existed in the original message).

The /RETURN_ADDRESS=address qualifier can be used to supply an alternate list return address when /REPLY_TO=LIST is specified.

/[NO]REQUEST_CONFIRMATION[=INTERVAL=delta-time]

Controls whether or not subscription requests should be confirmed. When /REQUEST_CONFIRMATION is specified, SUBSCRIBE and ADD requests will generate a special subscription confirmation message that is sent to the potential subscriber. If the user replies within the timeout interval, the subscription process is completed normally. Otherwise, the subscription request is dropped. Confirmations are disabled by default.

The default timeout interval for confirmation requests is three days (72 hours). You can change the timeout by specifying the INTERVAL keyword with a VMS delta-time value.

/[NO]RETURN_ADDRESS=address

This qualifier is used to specify an alternate address to be used as the ''Reply-To:'' address when /REPLY_TO=LIST is specified. This qualifier is most useful when multiple lists should have a common return address. For example, it can be used to redirect replies to a ''-Digest'' list back to the non-digest address.

/SETTINGS=(keyword[,...])

The /SETTINGS qualifier is used to override the default subscriber settings for a list. The valid keywords are MAIL, REPRO, CONCEAL, DIGEST, POST, and their "NO" forms.

A special keyword, DEFAULT, can be used to reset the settings to the MLF default for a mailing list. The default settings for a list are MAIL, REPRO, NOCONCEAL, NODIGEST, and POST.

/STRIP_HEADERS=keyword

This qualifier is used to strip certain RFC822 headers from messages posted to a mailing list.

The following keywords are supported:

- RECEIVED and NORECEIVED
- OTHER and NOOTHER

When /STRIP_HEADERS=RECEIVED is set, the ''Received:'' headers are stripped from the incoming message before it is mailed out to the list subscribers, thereby reducing the total number of ''Received:'' headers in the final message.

When /STRIP_HEADERS=OTHER is set, all ''other'' headers are stripped from posts. The ''other'' headers are any headers not recognized by MX, which includes such headers as X- headers, return-receipt headers, X.400 headers, etc. Setting a list to /STRIP_HEADERS=OTHER handily gets around potential problems with subscribers using the DOS package Pegasus Mail, which will send message receipt messages back to a list. Note that this may not be a viable setting for a mailing list that is gatewayed to a newsgroup, depending on the gateway software, since headers used by the gateway may be omitted.

See *Message Exchange Mailing List/File Server Guide* for more information on mailing list headers.

/[NO]SUBJECT_PREFIX="string"

The /SUBJECT_PREFIX qualifier can be used to add a prefix to the Subject line of messages posted to the list. By default, no prefix is added. When the list is set to /REPLY_TO=(SENDER), a short prefix string may be supplied to help subscribers recognize mailing list messages. The given string is bracketed by square brackets ([ ]) when it is prefixed to the subject lines. The maximum length for the prefix string is 32 characters. Prefix strings should be kept short to avoid generating extremely long subject lines.

/[NO]TEXT_ONLY

Controls whether or not the mailing list processor should accept only plain-text messages for list postings. If /TEXT_ONLY is specified, the mailing list processor examines the Content-Type: header on all list postings, if present, and ensures that the specified content type is ‘‘text/plain’’. Messages with no Content-Type: header are assumed to be plain-text. Rejected postings are returned to sender with an error message indicating that the list is configured to accept text messages only.

By default, no content type checking is performed on list postings.

/[NO]XHEADERS=("string"[,...])

The /XHEADERS qualifier can be used to add additional site-specific headers to mailing list posts. For example, you can use /XHEADERS to add additional non-standard "X-List-" headers such as "X-List-Archives". The format of the header string is: "Keyword: text". For example, "Precedence: Bulk", which is a non-standard header used by some mailers.

Extreme care should be taken when adding additional headers to mailing lists to ensure that duplicate headers or improperly formatted headers (those that don’t comply with RFC 822) aren’t added to mailing list posts.

See *Message Exchange Mailing List/File Server Guide* for more information on mailing list headers.

DEFINE LOCAL_DOMAIN

Defines a host name or wildcard pattern as a "local" domain for the SMTP server.

FORMAT

DEFINE LOCAL_DOMAIN *name-or-pattern*

PARAMETERS

name-or-pattern
A host name or a string containing VMS-style wildcard characters against which an SMTP host name should be matched.

DESCRIPTION

This command is used in conjunction with the SMTP server's /NORELAY setting to establish domains that the SMTP server considers local, to prevent messages to or from these domains from being refused. When relaying in the SMTP server is disabled, it refuses to deliver messages to remote mailboxes when the originating mailbox is also remote. By default, the SMTP server considers any mailbox resolving to a LOCAL path, or with a host name sharing the same parent domain as the MX host name or TCP/IP host name, as being local. If your system is acting as a gateway for hosts in other domains, and you disable SMTP relaying, you should list the names of those other hosts, or a wildcard pattern that will match those names, in a DEFINE LOCAL_DOMAIN command, to ensure that messages coming from or going to those hosts will not be refused.

You may specify multiple local domains by using multiple DEFINE LOCAL_DOMAIN commands. Only one name or wildcard pattern is accepted per command. Use the REMOVE LOCAL_DOMAIN command to delete domains from the local-domain list.

DEFINE PATH

Defines a mapping between a domain name and a distribution path.

FORMAT

DEFINE PATH *domain-name path-name*

Command Qualifiers	Defaults
/ROUTE=host-name[@port-number]	

PARAMETERS

domain-name

A domain name or pattern containing VMS wildcards.

path-name

One of the supported MX path names: LOCAL, SMTP, SITE, DECNET_SMTP, or HOLDING_QUEUE=*n*.

DESCRIPTION

This command is used to associate a domain name and a distribution path. The Router uses this information to determine which distribution path should be used when routing mail messages. Each DEFINE PATH command adds a path definition to the list. The list is automatically sorted based on the length of the path and the presence of wildcards. The Router searches this list until the domain name of the address it is trying to route to matches the domain name or wildcard pattern of the path definition.

QUALIFIERS

/ROUTE=host-name[@port-number]

Specifies the name of a host that will route messages for the specified domain. For SMTP and HOLDING_QUEUE paths, this host name *must* directly resolve to an IP address; it cannot be the name that has only MX information in the Domain Name System.

For SMTP and HOLDING_QUEUE paths, you may specify a TCP port number to be used instead of the default SMTP port (25) when connecting to the specified host by appending an at-sign ''@'' and the port number (as a decimal integer) to the host name on this qualifier.

DEFINE REWRITE_RULE

Defines an address-rewriting rule for use by the Router.

FORMAT

DEFINE REWRITE_RULE *pattern result*

Command Qualifiers	Defaults
/REGEX	

PARAMETERS

pattern

An RFC 821-compliant address string, possibly with the addition of one or more substitution strings, or a regular expression (if the /REGEX qualifier is specified). The address string must include the opening and closing angle brackets. Any address matching *pattern* will be rewritten by the Router based on the *result* string.

result

An RFC 821-compliant address string, possibly with the addition of one or more substitution strings or (if the /REGEX qualifier is specified) subexpression references.

DESCRIPTION

This command is used to provide the Router with rules for transforming some addresses into other forms. The *pattern* string is an address string that must be matched to have the transformation apply. For example:

```
MCP> DEFINE REWRITE_RULE "<{user}@{host}.DECnet.mycompany.com>" -
_MCP>                    "<""{host}::{user}""@myhost.mycompany.com>"
```

The strings ‘‘{user}’’ and ‘‘{host}’’ are called *substitution strings*. They are identified by the curly braces surrounding the substitution name, which you may specify arbitrarily. In the *pattern* string, a substitution string matches any number of any characters, like the asterisk in a VMS wildcard pattern. The matched string can be substituted into the rewritten address by specifying the same substitution string in the *result* string, or it may be omitted.

Rewriting rules can be used when the DEFINE PATH/ROUTE command is inadequate, such as when a message must pass through two or more gateways to get to its destination, or when the rewrite affects both the local-part and the domain-part of an address. They should be used sparingly, however, since every address must be matched against the rewrite rules list.

The rewrite rules list is searched in the order you specify, so you should place more specific rules before more general rules. For non-regular-expression rules, pattern matching is done from right to left.

QUALIFIERS

/REGEX

Specifies that the rewrite rule uses UNIX-style regular expression matching instead of substitution strings. Regular expressions provide more sophisticated matching capabilities; see Appendix A for more information. When using regular expressions, substitutions are performed by enclosing matching subexpressions in parentheses in the pattern string and referencing those subexpressions with ''*n*'' in the result string.

For example, the rewrite rule mentioned in the Description section above could be written using regular expressions as:

```
MCP> DEFINE REWRITE_RULE/REGEX "<([[:username:]]+)@([[:domain:]]+)\.DECnet\.mycompa
_MCP>                          "<""\2::\1""@myhost.mycompany.com>"
```

Regular expressions use a syntax that can be difficult to understand. MCP will flag regular expressions that do not compile properly, but you should be careful to test your rewrite rules to ensure that they execute properly.

DEFINE SYSTEM_USERS

Defines the address to be given SYSTEM access to mailing lists.

FORMAT

DEFINE SYSTEM_USERS *address[,...]*

PARAMETERS

address[,...]

One or more addresses, separated by commas. Each of the users identified by these addresses will be considered ‘‘system’’ users by the mailing list processor, and granted access via the SYSTEM protection class to all mailing lists. Case is important only in the username portion of the address. To retain the case of the address, surround it with quotation marks.

DESCRIPTION

This command is used to provide the mailing list processor with a list of privileged users. These users are granted access to mailing lists via the SYSTEM protection class, and are also given CONTROL access to all mailing lists. They receive all messages sent to MXserver that cannot be handled automatically by the mailing list processor.

The first address on the SYSTEM_USER list is used as the return address for generic MXserver replies (those replies that are not about a specific mailing list). For this reason, you may want to specify an alias as the first system user.

Typically only the system manager and/or postmaster for the system should be identified as system users. This will allow them to control a mailing list on the system when the owners of the list cannot be contacted.

EXIT

Exits MCP.

FORMAT

EXIT

DESCRIPTION

Use this command to leave MCP. If you have modified the MX configuration, it is saved before exiting. If the configuration file has not been named, you are prompted for a file name before exiting.

HELP

Displays help information.

FORMAT

HELP *[topic...]*

PARAMETERS

topic
The name of a topic in the help library. If omitted, a list of topics is displayed.

MODIFY

Modifies existing configuration information.

FORMAT

MODIFY { *ALIAS alias new-fwdaddr* | *FILE_SERVER fsrv-name* | *LIST list-name* | *PATH domain new-path* | *REWRITE_RULE pattern new-result* }

DESCRIPTION

This command alters configuration information of the types listed in above. Each of the MODIFY commands takes the same arguments and qualifiers as its corresponding DEFINE command, so refer to the appropriate DEFINE command for further information.

MODIFY USER

Modifies a user's password in the SMTP authentication database.

FORMAT

MODIFY USER *username*

Command Qualifiers	Defaults
/PASSWORD=password-text	*/PASSWORD="PASSWORD"*

PARAMETERS

username
The username whose password is to be modified. Usernames may be up to 16 characters in length, and are case-sensitive. To specify a lowercase or mixed-case username, surround it with quotation marks.

DESCRIPTION

This command modifies an entry in the SMTP authentication database.

QUALIFIERS

/PASSWORD=password-text
Specifies a password for this username. Passwords can contain any characters, and are case-sensitive. They may be up to 64 characters in length. assigned.

QUEUE CANCEL

Cancels one or more queue entries.

FORMAT

QUEUE CANCEL *[entry-number,...]*

Command Qualifiers	Defaults
/[NO]LOG	*/NOLOG*

PARAMETERS

entry-number

Queue entry number to be cancelled. If the number of a base queue entry, all related agent-specific entries will also be cancelled.

If this parameter is omitted, all entries selected by the last QUEUE SELECT command are cancelled.

DESCRIPTION

This command sets the status of the specified queue entries to CANCELLED, which prevents further processing of the entries. This should only be done on entries which are not currently being processed by the Router or one of the delivery agents.

QUALIFIERS

/[NO]LOG

Causes a message to be displayed for each cancelled entry. The default is /NOLOG.

QUEUE COMPRESS

Compress the message queue file.

FORMAT

QUEUE COMPRESS

Command Qualifiers	Defaults
/MAXIMUM_ENTRIES=value	*None.*
/[NO]LOG	*/NOLOG*

DESCRIPTION

Shrinks the message queue file by creating a new file and renumbering all the existing entries in the file. This command may be used to create a smaller message queue, which affects the maximum number of entries allowed in the queue.

The /MAXIMUM_ENTRIES qualifier is required.

This command requires exclusive access to the MX message queue file. Before compressing the file, all MX agents must either be shut down or inactive.

QUALIFIERS

/MAXIMUM_ENTRIES=number-of-entries

Specifies the maximum number of queue entries to be allowed. MX will not allow more entries to be added to the queue than the specified value. MCP QUEUE EXTEND can be used to increase the number of allowed entries.

The size of the queue file in blocks is equal to the maximum number of entries, plus 10 blocks, plus whatever is added for the disk cluster.

/[NO]LOG

Causes a status message to be displayed after successful operation. Default is /NOLOG.

QUEUE CREATE

Create a message queue file.

FORMAT

QUEUE CREATE *[filespec]*

Command Qualifiers	Defaults
/MAXIMUM_ENTRIES=value	*None.*

PARAMETERS

filespec

Name of the queue control file to be created. If omitted, the default name, MX_FLQ_DIR:MX_SYSTEM_QUEUE.FLQ_CTL, is used.

DESCRIPTION

Creates a new, empty MX message queue file. The /MAXIMUM_ENTRIES qualifier is required.

Note: This command simply creates a new queue file; the existing queue file is not automatically deleted. Any files for any existing queue entries are also left in place.

This command requires exclusive access to the MX message queue file. Before compressing the file, all MX agents must either be shut down or inactive.

QUALIFIERS

/MAXIMUM_ENTRIES=number-of-entries

Specifies the maximum number of queue entries to be allowed. MX will not allow more entries to be added to the queue than the specified value. MCP QUEUE EXTEND can be used to increase the number of allowed entries.

The size of the queue file in blocks is equal to the maximum number of entries, plus 10 blocks, plus whatever is added for the disk cluster.

QUEUE DUMP

Dumps the contents and envelope of a message to a set of text files for editing.

FORMAT

QUEUE DUMP *entry-number*

Command Qualifiers	Defaults
/[NO]CANCEL	*/CANCEL*
/OUTPUT=file-spec	

PARAMETERS

entry-number

Number of the queue entry to be dumped.

DESCRIPTION

This command creates a set of files that can be used to modify message headers and/or recipient addresses for any pending (IN-PROGRESS or READY) message currently in the queue.

The output files are suitable for use with the MX_SITE_IN program so the message can be requeued after being dumped. By default, the three files are placed in the current default directory, named as shown in Table MCP–5.

Table MCP–5 QUEUE DUMP Output Files

File name	Description
ENTRY_*n*.RECIPIENTS	List of recipient addresses, one per line.
ENTRY_*n*.MSG_TEXT	Full text of message, including RFC822 headers.
REQUEUE_*n*.COM	Command procedure for invoking MX_SITE_IN to requeue the message.

You can modify the directory and/or base filename of the output files with the /OUTPUT qualifier.

QUALIFIERS

/[NO]CANCEL

Specifies whether or not the dumped queue entry should be cancelled. The dumped entry is automatically cancelled on successful completion of a QUEUE DUMP command, unless /NOCANCEL is specified.

/OUTPUT=file-spec

Specifies the device, directory, and/or base filename for the output files. The file type and version fields are not permitted with this qualifier. By

default, the output files are placed in the current directory, with names as shown in Table MCP–5.

QUEUE EXTEND

Extends the message queue file.

FORMAT

QUEUE EXTEND

Command Qualifiers	Defaults
/MAXIMUM_ENTRIES=value	*None.*

DESCRIPTION

Extends the existing message queue file to allow more entries to be in the queue at any given time.

The /MAXIMUM_ENTRIES qualifier is required.

This command requires exclusive access to the MX message queue file. Before compressing the file, all MX agents must either be shut down or inactive.

QUALIFIERS

/MAXIMUM_ENTRIES=number-of-entries

Specifies the maximum number of queue entries to be allowed. MX will not allow more entries to be added to the queue than the specified value. MCP QUEUE EXTEND can be used to increase the number of allowed entries.

The size of the queue file in blocks is equal to the maximum number of entries, plus 10 blocks, plus whatever is added for the disk cluster.

QUEUE HOLD

Places a queue entry on hold.

FORMAT

QUEUE HOLD *[entry-number,...]*

PARAMETERS

entry-number

Number of the queue entry to be held.

If this parameter is omitted, all entries selected by the last QUEUE SELECT command are held.

DESCRIPTION

THis command places a queue entry on hold, so it will not be processed. Use the QUEUE READY command to release the entry for processing.

QUEUE PURGE

Purges the message queue of finished and cancelled entries.

FORMAT

QUEUE PURGE

Command Qualifiers	Defaults
/[NO]LOG	*/NOLOG*

DESCRIPTION

This command searches the message queue for all entries of FINISH or CANCELLED status and deletes them from the queue.

QUALIFIERS

/[NO]LOG

Causes a message to be displayed for each deleted entry. The default is /NOLOG.

QUEUE READY

Readies a queue entry.

FORMAT

QUEUE READY *[entry-number,...]*

Command Qualifiers	Defaults
/AFTER=date-time	
/FINAL	
/[NO]LOG	*/NOLOG*

PARAMETERS

entry-number

Queue entry number to be readied. If the number of a base queue entry, the base entry will be readied and all existing agent-specific entries will be cancelled.

If this parameter is omitted, all entries selected by the last QUEUE SELECT command are held.

DESCRIPTION

This command sets the status of the specified queue entries to READY and clears the delay flag. This should only be done on entries which are not currently being processed by the Router or one of the delivery agents.

QUALIFIERS

/AFTER=date-time

Specifies a date and time after which the entry should be processed. Without this qualifier, the entry is readied for immediate processing.

/FINAL

Specifies that the entry should be readied for a final attempt by the delivery agent. This qualifier causes the retry counts to be set to their maximum value before the entry is readied. This will cause the agent to attempt to deliver the message just once; if the delivery fails, the message will be returned to sender.

Use this qualifier to force a message to be ‘‘bounced’’ when you know that delivery to the intended recipient(s) is impossible.

/[NO]LOG

Causes a message to be displayed for each readied entry. The default is /NOLOG.

QUEUE SELECT

Selects queue entries.

FORMAT

QUEUE SELECT

Command Qualifiers	Defaults
/BEFORE=time	
/CREATED	
/DELAY	
/DESTINATION_AGENT=agent	
/EXPIRE	
/HELD	
/IN_PROGRESS	
/MODIFIED	
/ORIGIN_AGENT=agent	
/SINCE=time	
/WAITING	

DESCRIPTION

This command builds a list of queue entries based on a combination of selection criteria that you specify. Subsequent QUEUE CANCEL, QUEUE HOLD, and QUEUE READY commands will use this selection list if you do not specify entry numbers on those commands.

You can display the selected queue entries with the QUEUE SHOW/SELECTED command.

QUALIFIERS

/BEFORE[=time]

Selects only those entries dated before the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the /BEFORE qualifier to indicate the time attribute to be used as the basis for selection: /CREATED (default), /DELAY, /EXPIRE, or /MODIFIED.

/CREATED

Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /CREATED qualifier selects entries based on their dates of creation.

/DELAY

Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /DELAY qualifier selects entries based on their delay dates.

/DESTINATION_AGENT=agent

Selects only those entries that are to be or have been processed by the specified MX agent. Valid keywords are: ROUTER, MLF, LOCAL, SMTP, SITE, DNSMTP, and HOLDING_QUEUE=*n*.

/EXPIRE
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /EXPIRE qualifier selects entries based on their dates of expiration.

/HELD
Selects only those entries that are in USER-HOLD or OPER-HOLD state.

/IN_PROGRESS
Displays only entries marked as being in-progress (INPROG).

/MODIFIED
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /MODIFIED qualifier selects entries based on their dates of modification.

/ORIGIN_AGENT=agent
Selects only those entries that were entered into the queue by the specified MX agent. Valid keywords are: LOCAL, SMTP, SITE, MAIL, and DNSMTP.

/SINCE[=time]
Selects only those entries dated after the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the /SINCE qualifier to indicate the time attribute to be used as the basis for selection: /CREATED (default), /DELAY, /EXPIRE, or /MODIFIED.

/WAITING
Selects only those entries with READY status.

QUEUE SHOW

Displays queue entries.

FORMAT

QUEUE SHOW *[entry-number,...]*

Command Qualifiers	Defaults
/ALL	
/BEFORE=time	
/BRIEF	
/CREATED	
/DATE	
/DELAY	
/DESTINATION_AGENT=agent	
/EXPIRE	
/FULL	
/HELD	
/IN_PROGRESS	
/MODIFIED	
/ORIGIN_AGENT=agent	
/OUTPUT=file-spec	
/SELECTED	
/SINCE=time	
/WAITING	

PARAMETERS

entry-number

Queue entry number to be displayed. If omitted, all READY and IN-PROGRESS entries are displayed.

DESCRIPTION

This command displays entries in the message queue.

QUALIFIERS

/ALL

Causes all queue entries to be displayed, regardless of status. If omitted, just the READY and IN-PROGRESS entries are displayed.

/BEFORE[=time]

Selects only those entries dated before the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the /BEFORE qualifier to indicate the time attribute to be used as the basis for selection: /CREATED (default), /DELAY, /EXPIRE, or /MODIFIED.

/BRIEF
Causes a brief listing of all the queue entries to be displayed, including those that have finished or been cancelled. The information displayed is taken only from the MX queue file and includes the target MX process for each entry.

/CREATED
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /CREATED qualifier selects entries based on their dates of creation.

/DATE
Causes the creation and modification dates to be displayed for each queue entry.

/DELAY
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /DELAY qualifier selects entries based on their delay dates.

/DESTINATION_AGENT=agent
Selects only those entries that are to be or have been processed by the specified MX agent. Valid keywords are: ROUTER, MLF, LOCAL, SMTP, SITE, DNSMTP, and HOLDING_QUEUE=*n*. This qualifier is most useful when used with /BRIEF.

/EXPIRE
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /EXPIRE qualifier selects entries based on their dates of expiration.

/FULL
Provides more details about the displayed entries, including intended recipients, error counts, and last error status messages. If omitted, a brief, one-line display is produced for each entry.

/HELD
Selects only those entries that are in USER-HOLD or OPER-HOLD state.

/IN_PROGRESS
Displays only entries marked as being in-progress (INPROG).

/MODIFIED
Modifies the time value specified with the /BEFORE or the /SINCE qualifier. The /MODIFIED qualifier selects entries based on their dates of modification.

/ORIGIN_AGENT=agent
Selects only those entries that were entered into the queue by the specified MX agent. Valid keywords are: LOCAL, SMTP, SITE, MAIL, and DNSMTP.

/OUTPUT=file-spec
Directs the results to the specified file. If omitted, the results are displayed on SYS$OUTPUT.

/SELECTED

Displays entries selected by the last QUEUE SELECT command. If this qualifier is specified, no other selection qualifiers or entry numbers may specified.

/SINCE[=time]

Selects only those entries dated after the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Specify one of the following qualifiers with the /SINCE qualifier to indicate the time attribute to be used as the basis for selection: /CREATED (default), /DELAY, /EXPIRE, or /MODIFIED.

/WAITING

Limits the display to only those entries with READY status.

QUEUE STATISTICS

Displays statistical information concerning the entries in the MX message queue.

FORMAT

QUEUE STATISTICS

DESCRIPTION

This command displays the total number of entries in the queue, the maximum number of entries possible for the queue file, the percentage of entries in use, and the largest entry number ever used during the life of the file.

QUEUE SYNCHRONIZE

Synchronizes the message queue bitmap with the actual entries in the queue.

FORMAT

QUEUE SYNCHRONIZE

Command Qualifiers	Defaults
/LOG	
/RESET	

DESCRIPTION

This command updates the bitmap for the MX system message queue to synchronize it with the actual entries in the queue. The only time this command may be necessary is in the event of a system crash or disk failure.

The command may be issued at any time; it does not require exclusive access to the MX message queue file.

QUALIFIERS

/LOG

Causes a status message to be displayed after successful operation. Default is /NOLOG.

/RESET

Resets the ‘‘Highest entry used’’ counter displayed by QUEUE STATISTICS. By default, the counter is not reset.

QUIT

Leaves MCP without saving any configuration changes.

FORMAT QUIT

DESCRIPTION

Use this command leave MCP without saving any of the changes made to the MX configuration. If the configuration was changed, MCP will ask for confirmation before returning to DCL.

REMOVE

Removes a configuration record.

FORMAT

REMOVE { *ALIAS alias* | *FILE_SERVER fileserv-name* | *INSIDE_NETWORK_ADDRESS ip-address* | *LIST list-name* | *LOCAL_DOMAIN name-or-pattern* | *PATH domain* | *REWRITE_RULE pattern* }

DESCRIPTION

This command removes one record of the specified type from the MX configuration. The specified alias, inside network address, list name, domain, rejection rule, or rewrite rule pattern must match exactly the identical field in the record to be removed.

When removing an inside network address, you must also specify the /NETMASK qualifier when the address is for a network rather than a host.

REMOVE USER

Removes an entry from the SMTP authentication database.

FORMAT

REMOVE USER *username*

PARAMETERS

username

The username to be removed from the database. Usernames may be up to 16 characters in length, and are case-sensitive. To specify a lowercase or mixed-case username, surround it with quotation marks.

DESCRIPTION

This command removes an entry in the SMTP authentication database. Once removed, an SMTP client can no longer user the username to authenticate to the SMTP server. The authentication sequence will be rejected, and any messages sent by the client will be subject to normal anti-relay and anti-junk-mail processing (if the client is on an ‘‘outside’’ network).

RESET

Sends a reset signal to one or more delivery agents.

FORMAT

RESET *[agent-name...]*

Command Qualifiers	Defaults
/ACCOUNTING	
/CLUSTER	*/CLUSTER*
/NODE[=(node,...)]	*/CLUSTER*

PARAMETERS

agent-name...

One or more MX delivery agent names, separated by commas. Valid names are DECNET_SMTP, LOCAL, MLF, ROUTER, SITE, SMTP, and HOLDING_QUEUE=(*n*,...). If omitted, all agents running on the same node as the user executing this command are reset.

DESCRIPTION

The RESET command can be used to signal one or more MX delivery agents to reload their configuration information. **This command requires the SYSLCK privilege.**

QUALIFIERS

/ACCOUNTING

Causes the specified agents to open new versions of their accounting files. Only useful for those agents that support accounting, and with MLF (which causes a new version of FILESERV_LOG.LOG to be opened).

If /ACCOUNTING is specified, no reload of configuration data is performed; only the accounting files are reset.

/CLUSTER

Specifies that the RESET command should affect all of the specified agents cluster-wide, rather than just the ones on the node from which the command is executed. This is the default behavior for RESET; use /NODE to have the reset affect agents on specific nodes in the cluster.

/NODE[=(node,...)]

Specifies that the RESET command should affect only the specified agents running on the given nodes. If no node names are specified, the local node is used by default. If this qualifier is omitted, all agents cluster-wide are reset.

REVIEW

Displays the subscribers of a local mailing list.

FORMAT

REVIEW *mailing-list*

Command Qualifiers	Defaults
/OUTPUT=file-spec	

PARAMETERS

mailing-list

Name of the mailing list whose subscriber list is to be displayed. The mailing list must reside on the local system.

DESCRIPTION

This command performs the functional equivalent of the mailing list processor's REVIEW command for any mailing list on the local system. All subscribers' addresses and personal names (if any) listed, along with their MAIL/NOMAIL status.

QUALIFIERS

/OUTPUT=file-spec

Directs the results to the specified file. If omitted, the results are displayed on SYS$OUTPUT.

SAVE

Saves the current configuration to a file.

FORMAT

SAVE *file-spec*

PARAMETERS

file-spec
The name of the file to which the configuration is written. If omitted, the file type defaults to MXCFG.

DESCRIPTION

Use this command to write the MX configuration you are creating or changing to a file. You should save the configuration to the file MX_DIR:MX_CONFIG.MXCFG if you want it to be used by the MX processing agents.

SET DECNET_SMTP

Alters settings specific to the SMTP-over-DECnet delivery agent.

FORMAT

SET DECNET_SMTP

Command Qualifiers	Defaults
/[NO]ACCOUNTING	
/MAXIMUM_RETRIES=count	
/RETRY_INTERVAL=delta-time	

DESCRIPTION

This command is used to change the SMTP-over-DECnet agent settings.

QUALIFIERS

/[NO]ACCOUNTING

Enables or disables the recording of accounting information. Accounting is disabled by default. When enabled, accounting information is written to the file MX_DNSMTP_DIR:MX_DNSMTP_ACC.DAT. You can redirect the accounting information to another file by defining the logical name MX_DNSMTP_ACC.

The format of the accounting record is:

```
dd-mmm-yyyy hh:mm XMIT: PROTO=DECNET_SMTP, SOURCE="src-addr", HOST="host", BYTES_SENT=n
```

where **dd-mmm-yyyy hh:mm** is the date/time stamp of the accounting record; **src-addr** is the source address for the message; **host** is the host to which the message was sent; and **n** is the number of bytes in the delivered message.

/MAXIMUM_RETRIES=count

Sets the maximum number of retries for message delivery. The default count is 96, which for a half-hour retry interval comes to roughly two days of retries.

/RETRY_INTERVAL=delta-time

Sets the amount of time that should elapse between delivery attempts. The default is 30 minutes. Specify as a VMS delta time value.

SET LOCAL

Alters Local-delivery-specific settings.

FORMAT SET LOCAL

Command Qualifiers	Defaults
/[NO]ACCOUNTING	*/NOACCOUNTING*
/[NO]CC_POSTMASTER	*/NOCC_POSTMASTER*
/[NO]DISABLE_EXQUOTA[=FATAL]	*/NODISABLE_EXQUOTA*
/[NO]HEADERS=(loc:(hdrname[,...])[,...])	
/[NO]LONG_LINES	*/NOLONG_LINES*
/MAXIMUM_RETRIES=count	*See text*
/[NO]MULTIPLE_FROM	*/MULTIPLE_FROM*
/[NO]OMIT_RESENT_HEADERS	*/NOOMIT_RESENT_HEADERS*
/[NO]QP_DECODE	*/QP_DECODE*
/RETRY_INTERVAL=delta-time	*See text*

DESCRIPTION

This command is used to change the local delivery agent settings.

QUALIFIERS

/[NO]ACCOUNTING

Enables or disables the recording of accounting information. Accounting is disabled by default. When enabled, accounting information is written to the file MX_LOCAL_DIR:MX_LOCAL_ACC.DAT. You can redirect the accounting information to another file by defining the logical name MX_ LOCAL_ACC.

The format of the accounting record is:

```
dd-mmm-yyyy hh:mm DELIVER: SOURCE="src-addr", USER="user", SIZE=n
```

where **dd-mmm-yyyy hh:mm** is the date/time stamp of the accounting record; **src-addr** is the source address for the message; **user** is the address on the local host to which the message was delivered; and **n** is the number of bytes in the delivered message.

/[NO]CC_POSTMASTER

Specifies whether or not error messages resulting from LOCAL delivery errors are mailed to the local POSTMASTER, in addition to the original message sender.

/[NO]DISABLE_EXQUOTA[=FATAL]

Specifies whether the local delivery agent should disable the EXQUOTA privilege when attempting to deliver messages to VMS MAIL. By default, EXQUOTA privilege remains enabled during local message delivery. Using the /DISABLE_EXQUOTA qualifier will prevent users' mailboxes from exceeding their diskquotas. In addition, specifying the FATAL keyword on this qualifier will cause MX to treat an over-quota condition as a non-recoverable error and immediately return any messages causing the

condition to sender. If FATAL is not specified, over-quota conditions are treated as transient errors and will be retried (subject to the /RETRY_ INTERVAL and /MAXIMUM_RETRIES settings).

/HEADERS=(loc:(hdrname[,...])[,...])

Controls the inclusion and placement of RFC 822 headers in messages delivered to VMS Mail. Valid values for **loc** are TOP and BOTTOM. Valid values for **hdrname** are listed in Table MCP–6.

Table MCP–6 Header name keywords

Keyword	Meaning
ALL	All headers.
BCC	The Bcc (blind carbon copy) header.
CC	The CC (carbon copy) header.
DATE	The Date header.
FROM	The From header.
IN_REPLY_TO	The In-Reply-To header.
KEYWORDS	The Keywords header (not strictly RFC 822).
MESSAGE_ID	The Message-Id header.
OTHER	Any header not recognized by MX.
RECEIVED	The Received header(s).
REFERENCES	The References header (not strictly RFC 822).
REPLY_TO	The Reply-To header.
RESENT_BCC	The Resent-Bcc header.
RESENT_CC	The Resent-CC header.
RESENT_DATE	The Resent-Date header.
RESENT_FROM	The Resent-From header.
RESENT_MESSAGE_ID	The Resent-Message-Id header.
RESENT_REPLY_TO	The Resent-Reply-To header.
RESENT_SENDER	The Resent-Sender header.
RESENT_TO	The Resent-To header.
RETURN_PATH	The Return-Path header.
SENDER	The Sender header.
SUBJECT	The Subject header.
TO	The To header.

The header names can be negated by prefixing them with NO. You may include any combination of headers for inclusion at the top and/or the end of the message. For example, you might want to move the Received and Return-Path headers to the bottom of messages, since the information they convey are of interest only when there are network problems:

```
MCP> SET LOCAL/HEADERS=(TOP:(ALL,NORECEIVED,NORETURN_PATH),-
_MCP>                   BOTTOM:(NOALL,RECEIVED,RETURN_PATH))
```

/[NO]LONG_LINES

Enables support for delivery of messages containing lines longer than the MAIL-11 maximum of 511 characters for non-DECnet addresses. When enabled, the local delivery agent will only truncate lines at 511 characters when the destination address contains a colon (indicating that the recipient is a DECnet address). Disabled by default.

Note: When /LONG_LINES is enabled, delivery to DECnet addresses through VMS MAIL aliases (either in the forwarding database or through logical names) may fail because MX cannot determine that such aliased addresses should be treated as DECnet deliveries rather than local deliveries.

/MAXIMUM_RETRIES=count

Sets the maximum number of retries for DECnet message delivery. The default count is 96, which for a half-hour retry interval comes to roughly two days of retries.

/[NO]MULTIPLE_FROM

Controls whether or not the VMS Mail ''From:'' line on incoming messages can contain multiple return addresses. By default, if an RFC822 From: or Reply-To: line contains more than one address, as many of those addresses as will fit are included on the VMS Mail ''From:'' line (up to 255 characters). Specifying /NOMULTIPLE_FROM limits the ''From:'' line to a single address.

/[NO]OMIT_RESENT_HEADERS

Controls whether or not Resent- headers should be omitted when the local delivery agent performs forwarding due to a VMS MAIL forwarding database entry containing an MX% address. By default, Resent- headers are included; this helps MX detect forwarding loops. However, some PC e-mail clients treat Resent- headers in a confusing way, so omitting them may be desirable.

/[NO]QP_DECODE

Controls whether or not incoming MIME quoted-printable messages are automatically decoded by MX Local before delivery through VMS Mail. By default, such messages are decoded. If your users read their mail via POP or IMAP, you might want to disable the decoding to let the users' browsers do the decoding.

/RETRY_INTERVAL=delta-time

Sets the amount of time that should elapse between delivery attempts. The default is 30 minutes. Specify as a VMS delta time value.

SET MLF

Alters MLF (Mailing List/File server) settings.

FORMAT

SET MLF

Command Qualifiers	Defaults
/[NO]DELAY_DAYS[=(dow...)]	
/[NO]RECIPIENT_MAXIMUM[=n]	

DESCRIPTION

This command sets global parameters for the Mailing List/File Server agent (MLF).

QUALIFIERS

/[NO]DELAY_DAYS[=(dow...)]

Specifies the days of the week on which file servers' send-delay threshold should be respected. The default is to respect send-delay thresholds on every day of the week. A list of day-of-week names may be specified. Specify /NODELAY_DAYS to globally disable send-delay thresholds on all file servers.

/[NO]RECIPIENT_MAXIMUM[=n]

Sets the maximum number of recipients per message generated by the MLF agent. If your MLF agent services large mailing lists with many remote subscribers, you may want to use this setting to limit the number of recipients per message generated by MLF. This will break up the distribution to the mailing list into smaller chunks, allowing for more parallelism in delivery.

Setting too small a value, however, could create a lengthy backlog in your MX message queue, depending on the number of subscribers on your mailing list(s) and the number of messages the list receives each day.

The default is /NORECIPIENT_MAXIMUM, which forces each incoming mailing list message to be forwarded as just one outbound message.

SET ROUTER

Alters Router-specific settings.

FORMAT SET ROUTER

Command Qualifiers	Defaults
/[NO]ACCOUNTING	
/[NO]OMIT_VMSMAIL_SENDER	
/[NO]PERCENT_HACK	

DESCRIPTION

This command is used to enable or disable the automatic de-hacking of percent signs in a local address. Percent-hacking is explained in Section 3.3.

QUALIFIERS

/[NO]ACCOUNTING

Enables or disables the recording of accounting information. Accounting is disabled by default. When enabled, accounting information is written to the file MX_ROUTER_DIR:MX_ROUTER_ACC.DAT. You can redirect the accounting information to another file by defining the logical name MX_ROUTER_ACC.

The format of the accounting record is one of the following:

```
dd-mmm-yyyy hh:mm ROUTE: SENDER="src-addr", RCPT="rcp-addr", ROUTE="host", PATH=path, BYTES=n
dd-mmm-yyyy hh:mm BOUNCE: SENDER="src-addr", RCPT="rcp-addr", ORIGRCPT="orcp-addr", ERROR="errmsg"
```

where **dd-mmm-yyyy hh:mm** is the date/time stamp of the accounting record; **src-addr** is the source address for the message; **rcp-addr** is the recipient's address; **orcp-addr** is the recipient's address before it was changed by any rewrite rules; **host** is the host route associated with the matching path; **path** is the delivery path (agent) that will deliver the message; **errmsg** is the text of the error message describing why routing failed; and **n** is the number of bytes in the delivered message.

/[NO]OMIT_VMSMAIL_SENDER

Enables or disables the omission of the Sender: header for messages sent from VMS Mail. Normally, a Sender: line is included if the Sender: and From: addresses are different. However, some sites using the MX_SITE_NAME_CONVERSION feature with the FULL_CONVERT routine have had problems sending mail to some mailers when the Sender: and From: are different, despite the fact that it is allowed by RFC822 (in fact, according to RFC822, the Sender: should be omitted if it is the same address as the From: address). To allow those sites to work around the problems with those mailers, /OMIT_VMSMAIL_SENDER can be used to omit the Sender: line in all cases.

MX_SITE_NAME_CONVERSION is documented in the *Message Exchange Programmer's Guide*.

Note: If /OMIT_VMSMAIL_SENDER is specified, then the Sender: line is also omitted from any SMTP messages forwarded by users with the MX_FAKE_RFC822 process rights identifier.

/[NO]PERCENT_HACK

Enables or disables automatic percent-hack translation. The default is to enable translation.

SET SITE

Alters settings specific to the SITE delivery agent.

FORMAT

SET SITE

Command Qualifiers	Defaults
/MAXIMUM_RETRIES=count	
/RETRY_INTERVAL=delta-time	

DESCRIPTION

This command is used to change the SITE agent settings.

QUALIFIERS

/MAXIMUM_RETRIES=count

Sets the maximum number of retries for message delivery. The default count is 96, which for a half-hour retry interval comes to roughly two days of retries.

/RETRY_INTERVAL=delta-time

Sets the amount of time that should elapse between delivery attempts. The default is 30 minutes. Specify as a VMS delta time value.

SET SMTP

Alters SMTP-delivery-specific settings.

FORMAT SET SMTP

Command Qualifiers	Defaults
/[NO]ACCOUNTING	
/[NO]AUTHENTICATION=(type,...)	*/NOAUTHENTICATION*
/DEFAULT_ROUTER=hostname	
/DNS_RETRIES=dnscount	
/MAXIMUM_RETRIES=count	
/[NO]PERCENT_HACK	*/PERCENT_HACK*
/[NO]RBL_CHECK[=(domain,...)]	
/[NO]RELAY_ALLOWED	*/RELAY_ALLOWED*
/RETRY_INTERVAL=delta-time	
/[NO]VALIDATE_SENDER_DOMAIN	
/[NO]VERIFY_ALLOWED	

DESCRIPTION

This command is used to change the SMTP interface settings.

QUALIFIERS

/[NO]ACCOUNTING

Enables or disables the recording of accounting information. Accounting is disabled by default. When enabled, accounting information is written to the file MX_SMTP_DIR:MX_SMTP_ACC.DAT. You can redirect the accounting information to another file by defining the logical name MX_SMTP_ACC.

The format of the accounting record is:

```
dd-mmm-yyyy hh:mm XMIT: PROTO=SMTP, SOURCE="src-addr", HOST="dest", SENT-TO="relay", BYTES_SENT=n, RCPT=rcp
```

where **src-addr** is the source address for the message; **dest** is the name of the Internet host to which the message was directed; **relay** is the name of the Internet host to which the message was actually sent; **n** is the number of bytes transmitted; and **rcp-addr** is the recipient's address. One accounting message is written for each successful delivery of a message, so one message could result in multiple accounting records.

/[NO]AUTHENTICATION=(type,...)

Enables or disables the authentication extension in the SMTP server. The default is /NOAUTHENTICATION; when turned off, the extension is not advertised by the server to any clients. Supported values for *type* are CRAM_MD5 and PLAIN.

CRAM_MD5 enables the use of the CRAM-MD5 authentication mechanism, using the private authentication database maintained through the CREATE USER_DATABASE_FILE, ADD USER, and REMOVE USER commands in MCP.

PLAIN enables the use of the PLAIN and LOGIN authentication mechanisms, using the VMS SYSUAF as the authentication source for usernames and passwords.

/DEFAULT_ROUTER=hostname

Specifies the name of a host to which SMTP messages can be forwarded if MX cannot resolve a host name. This qualifier should only be used if you are not using the Internet domain name service. The **hostname** should be the name of a host which appears in your local host table.

/DNS_RETRIES=dnscount

Sets the maximum number of retries for SMTP delivery when the cause of the failure is the inability to determine the address corresponding to a host name. Certain types of domain server failures can cause MX to believe a host name is invalid. When a host name is genuinely invalid, however, the sender should be told relatively quickly. Therefore, the default count is 12 (giving about 6 hours' worth of attempts for a half-hour retry interval).

/MAXIMUM_RETRIES=count

Sets the maximum number of retries for SMTP message delivery. The default count is 96, which for a half-hour retry interval comes to roughly two days of retries.

/[NO]PERCENT_HACK

Controls whether the SMTP server accepts or rejects recipient addresses containing a percent sign in the username portion of the address (sometimes used to route messages through other systems). By default, such addresses are accepted. Unlike SET ROUTER/NOPERCENT_HACK, which disables the percent-hack only for local addresses, SET SMTP/NOPERCENT_HACK causes the SMTP server to reject *all* addresses with a percent sign in the username part of the address, regardless of whether or not the hostname part of the address is local or remote.

/[NO]RBL_CHECK[=(domain,...)]

Controls whether the SMTP server checks to see if a connecting host is on one or more Internet Realtime Blackhole Lists (RBLs) before allowing it to start an SMTP session. This is disabled by default. You must specify at least one RBL domain when enabling this check. Multiple domains may be specified.

See Section 8.5 for more information about the Realtime Blackhole List.

/[NO]RELAY_ALLOWED

Controls whether the SMTP server accepts ''relayed'' messages. A relayed message is a message from a remote source that lists recipients that are also remote. Relaying is allowed by default.

When relaying is disabled, the SMTP server examines the originator's address on each incoming message as well as each recipient's address. If both addresses are remote, delivery to the remote recipient(s) is refused. By default, any address that maps to a LOCAL path is considered local, as is any host name that shares the same parent domain as the MX host name or the TCP/IP host name. You may specify other host or domain names that should also be considered local with the DEFINE LOCAL_DOMAIN command.

/RETRY_INTERVAL=delta-time

Sets the amount of time that should elapse between delivery attempts. The default is 30 minutes. Specify as a VMS delta time value.

/[NO]VALIDATE_SENDER_DOMAIN

Enables or disables a check by the SMTP server on the validity of the domain name appearing in the SMTP MAIL FROM: command on incoming messages. When enabled, the SMTP server attempts to look up the domain name of the sender in the domain name system; if the name does not appear in the DNS (with any type of resource record), the MAIL FROM command, and hence the incoming message, will is rejected. This setting is disabled by default.

Enabling this option can be useful in preventing unwanted junk e-mail from entering your system, since junk e-mail is often sent using fictitious return addresses, sometimes with nonexistent domain names.

/[NO]VERIFY_ALLOWED

Enables or disables the processing of VRFY commands in the SMTP server. The VRFY command, a required command in the SMTP protocol, is often used by users on other systems to check whether or not a particular mailbox is valid. This information can be regarded as a security risk, and security-conscious system administrators may want to prevent VRFY commands from being processed.

By default, VRFY commands are allowed.

SHOW

Displays all or part of the current MX configuration.

FORMAT

SHOW { *ALIASES [pattern]* | *CONFIGURATION_FILE* | *DECNET_SMTP* | *FILE_SERVER [pattern]* | *KEY [key]* | *LISTS [pattern]* | *LOCAL* | *PATHS [pattern]* | *REWRITE_RULES [pattern]* | *ROUTER* | *SITE* | *SMTP* | *SYSTEM_USERS* | *USERS [pattern]* | *VERSION* | *ALL* }

Command Qualifiers	Defaults
/BRIEF	*/FULL*
/[NO]COMMAND	*/NOCOMMAND*
/FULL	*/FULL*
/OUTPUT=file-spec	*/OUTPUT=SYS$OUTPUT:*

DESCRIPTION

The SHOW command displays the specified parts of the current MX configuration. For aliases, file servers, lists, paths, and rewrite rules, only those records matching *pattern* (which may contain wildcard characters) are displayed; if you omit *pattern*, all records are displayed.

SHOW CONFIGURATION_FILE displays the name of the configuration file loaded when MCP was started. SHOW VERSION displays the version identifier for the current version of MX.

QUALIFIERS

/BRIEF

The /BRIEF qualifier indicates that the display should be formatted in an abbreviated format. The default is to display full information. This qualifier only affects SHOW LISTS output; the brief and full displays are identical for other SHOW commands.

/[NO]COMMAND

The /COMMAND qualifier indicates that the display should be formatted as the commands that would be entered to create the specified records. Use /COMMAND with the /OUTPUT qualifier to create an MCP command file that can be altered with your favorite editor, then read back into MCP to create a new configuration.

/FULL

The /FULL qualifier indicates that full information should be displayed. This is the default. This qualifier only affects SHOW LISTS output; the brief and full displays are identical for other SHOW commands.

/OUTPUT=file-spec

The /OUTPUT qualifier is used to direct the SHOW result to a file or other device. By default, the result is displayed on the current output device, SYS$OUTPUT.

SHUTDOWN

Sends a shutdown signal to one or more delivery agents.

FORMAT

SHUTDOWN *[agent-name...]*

Command Qualifiers	Defaults
/CLUSTER	*/NODE*
/NODE[=(node,...)]	*/NODE*
/[NO]WAIT	*/NOWAIT*

PARAMETERS

agent-name...

One or more MX delivery agent names, separated by commas. Valid names are DECNET_SMTP, LOCAL, MLF, ROUTER, SITE, SMTP, SMTP_SERVER, and HOLDING_QUEUE=(*n*,...). If omitted, all agents running on the same node as the user executing this command are shut down.

Note that the SMTP delivery agent may be shut down separately from the SMTP_SERVER message entry agent.

DESCRIPTION

The SHUTDOWN command can be used to signal one or more MX delivery agents to finish processing and exit. **This command requires the SYSLCK privilege.**

QUALIFIERS

/CLUSTER

Specifies that the SHUTDOWN command should affect the specified agents on all nodes in the cluster, not just the current node.

/NODE[=(node,...)]

Specifies that the SHUTDOWN command should affect the specified agents only on the given nodes. If /NODE is specified with no node names, agents on the local node are shut down; this is the default behavior.

/[NO]WAIT

Specifies whether or not MCP should wait until the agent(s) have responded to the shutdown notification before continuing. The default is /NOWAIT.

SPAWN

Spawns a subprocess.

FORMAT

SPAWN *[dcl-command]*

PARAMETERS

dcl-command

A DCL command to be executed in the subprocess. If omitted, MCP is suspended and the terminal is attached to the subprocess for interactive entry of commands. To return to the process running MCP, use the DCL LOGOUT command to end the subprocess.

DESCRIPTION

This command creates a subprocess to execute one or more DCL commands. See the description of the DCL SPAWN command for further information.

STATUS

Displays a list of the MX agent processes currently running and the current state of each agent process.

FORMAT

STATUS *[agent-name...]*

Command Qualifiers	Defaults
/NODE=(node[,...])	

PARAMETERS

agent-name...

One or more MX agent names, separated by commas. Valid names are DECNET_SMTP, LOCAL, MLF, ROUTER, SITE, SMTP, and SMTP_SERVER. If omitted, information about all agents is displayed.

DESCRIPTION

For each process running one of the MX agent programs, the process ID, process name, MX status, and MX agent type is displayed. In a VMScluster environment, the VMScluster node name for the process is also displayed. **This command requires the SYSLCK privilege.**

The status field indicates the action currently being performed by the agent. Valid status descriptions are shown in Table MCP–7.

Table MCP–7 MCP STATUS Descriptions

Unknown	Current status is not known.
Reading Config.	Reading the MX configuration file.
Idle	Process is idle, waiting for an entry to process.
Enabling	Requesting single agent enable lock.
Selecting	Searching in-memory queue for an entry to process.
Searching	Searching in-memory queue for an entry to process.
Locating	Initializing the in-memory queue by searching the MX queue file for entries to be processed by that agent.
Searching2	Searching in-memory queue for located entries.
Processing	Processing the specified entry.
Retrying	Retrying delivery of the specified entry.
Inserting	Inserting a new queue entry.
Search. for wait	Searching for delayed entries.
Waiting for	Idle, waiting to process the specified entry.
Req update	Requesting other agents to update an entry.
FLQ Cleanup	Performing MX queue file maintenance.
FLQread wait	Waiting for a read from the MX queue file.

Table MCP–7 (Cont.) MCP STATUS Descriptions

Wlock wait	Waiting for entry work lock.
Connected	SMTP Server has the specified number of incoming connections active.

QUALIFIERS

/NODE=(node[,...])

Specifies that the STATUS command should show only the specified agents running on the given nodes.

REJMAN Command Dictionary

REJMAN

Executes the MX Rejection Manager.

FORMAT

REJMAN *[command]*

Command Qualifiers	Defaults
/DATABASE=file-spec	

PARAMETERS

command

Any REJMAN command except the input redirection operator (@). The specified command is executed and control is returned to DCL immediately thereafter.

DESCRIPTION

REJMAN was written to be used as a DCL ''foreign'' command. To use it this way, you must define a symbol as follows:

```
$ REJMAN :== $MX_EXE:REJMAN
```

Defining the symbol in this way allows you to use the /DATABASE qualifier and specify ''one-host'' commands on the command line.

By default, REJMAN loads in the current configuration file, MX_DIR:MX_REJECTION_DATABASE.MXCFG, when started.

Note: REJMAN requires the SYSLCK privilege to be able to lock the rejection database from being modified by other processes.

QUALIFIERS

/DATABASE=file-spec

Loads the specified rejection database for editing. If not specified, the default database, MX_DIR:MX_REJECTION_DATABASE.MXCFG, is loaded.

Command Input Redirection

Execute REJMAN commands stored in a file.

FORMAT

@ *file-spec*

PARAMETERS

file-spec
Name of the file containing REJMAN commands. If the file type is omitted, the default type is MCP.

DESCRIPTION

Use this command to have REJMAN take further command input from the specified file. There is no built-in limit on the number of levels of nesting of command files, so be careful when using input redirection from within a command file.

This command can only be used at the REJMAN command prompt, not as a ''one-shot'' REJMAN command. To have a file be used for input for an entire REJMAN session, use the following sequence of DCL commands.

```
$ DEFINE/USER SYS$INPUT file-spec
$ REJMAN
```

ADD EXCLUSION

Adds an exclusion to heuristic junk mail filters.

FORMAT

ADD EXCLUSION *sender-pat*

Command Qualifiers	Defaults
/HEURISTIC=heuristic-name	

PARAMETERS

sender-pat

E-mail address or wildcard pattern to match the sender of the message, as it would appear in either the RFC822 From header or the SMTP MAIL FROM command.

DESCRIPTION

This command adds a new exception to one or all of the heuristic filters. If the sender (from either the RFC822 From header, or the SMTP MAIL FROM return path) of a message matches this address or pattern, the message is excluded from the specified heuristic filter check (or all checks if the /HEURISTIC qualifier is omitted).

QUALIFIERS

/HEURISTIC=heuristic-name

Specifies the heuristic filter to which this exclusion applies. If not specified, the exclusion is added to the global exclusion list for all heuristic filter checks.

ADD REJECTION

Adds a rejection rule to the database for the SMTP server.

FORMAT

ADD REJECTION *sender-pat [rcpt-pat]*

Command Qualifiers	Defaults
/ACCEPT[=accept-kwd]	
/ADDRESS=ip-addr-pat	
/FORWARD_TO=address	
/HEADER	
/MESSAGE=rejection-message	
/REGEX	

PARAMETERS

sender-pat

E-mail address or wildcard pattern to match the sender of the message, as it would appear in the SMTP MAIL FROM command. If the /HEADER qualifier is specified, this parameter is treated as a wildcard pattern to match an RFC822 header in the message, and the *rcpt-pat* parameter and other qualifiers are not used.

rcpt-pat

E-mail address or wildcard pattern to match the recipient of the message, as it would appear in the SMTP RCPT TO command. If omitted, matching only occurs against the sender, and a match causes the rejection to happen immediately after receipt of the MAIL FROM command, rather than on a per-recipient basis.

To match a character normally regarded as a wildcard (asterisk or percent sign), prefix the character with a backslash.

This parameter is not permitted if the /HEADER qualifier is specified.

DESCRIPTION

This command adds a new rejection rule to the database. For address-based rejection rules (those not specifying /HEADER), a match occurs for any of the following:

1 *sender-pat* is specified with no *rcpt-pat*, and the address in the SMTP MAIL FROM command matches the specified address or pattern. The sending system is notified of the rejection in the status code returned for the MAIL FROM command.

2 *sender-pat* and *rcpt-pat* are both specified. In this case, the SMTP MAIL FROM address must match *sender-pat* and the RCPT TO address must match the *rcpt-pat*. The check occurs on receipt of the RCPT TO command, and if a match occurs, the sending SMTP system is notified of the rejection in the status returned on the RCPT TO. Note that this may not prevent the delivery of the message to other users who do not match the *rcpt-pat* pattern.

In both cases, the rule can be restricted by the use of the /ADDRESS qualifier; if specified, the sending SMTP system's numeric IP address must match the address or pattern specified on that qualifier. You may modify the behavior of MX's SMTP server when a match occurs by using other qualifiers; see the qualifier descriptions for more information.

Header-based rules are specified with the /HEADER qualifier. Header-based rules are checked only after the entire message is received by the SMTP server, and if a match occurs, the sender is notified of the rejection after it transmits the termination sequence (CRLF-dot-CRLF) for the message. Header-based rules affect delivery to all specified recipients of a message; the *rcpt-pat* parameter is not used. Other qualifiers are not allowed when the /HEADER qualifier is used.

QUALIFIERS

/ACCEPT[=accept-kwd]

Specifies that the message should be accepted rather than rejected. This qualifier can be used to provide exceptions to other rejection rules; it applies only when both the *sender-pat* and *rcpt-pat* parameters are specified. Valid values for *accept-kwd* are:

- **AS_IS**. This is the default when no keyword is specified, and indicates that the recipient address should not be altered.
- **REWRITE=***new-rcpt-addr*. This indicates that the recipient's address should be rewritten to the specified new address. This mechanism can be used to re-direct messages that would otherwise be rejected to the postmaster or to another party that might be interested in tracking the unwanted messages.

This qualifier cannot be used in combination with the /MESSAGE qualifier or the /HEADER qualifier.

/ADDRESS=ip-addr-pat

Specifies either an IP address (in dotted-decimal form) or a wildcard pattern for matching an IP address. If specified, the IP address of the sending SMTP system is obtained for the incoming connection and matched against the specified address or pattern. This qualifier is not allowed when /HEADER is specified.

/FORWARD_TO=address

Specifies an e-mail address to which a message rejected by a header-based rule should be forwarded. This qualifier is used only with the /HEADER qualifier. When a message is forwarded using this qualifier, additional headers are included with the diverted message to indicate the original list of recipients for the message.

/HEADER

Modifies the type of rejection rule to be based on RFC822 headers, rather than on SMTP envelope addresses. When specified, the *sender-pat* parameter is used for matching RFC822 headers that might be present in the incoming message, and no other parameters or qualifiers are allowed.

/MESSAGE=rejection-message

Specifies the message emitted by the SMTP server which accompanies the SMTP status code indicating that the MAIL FROM or RCPT TO command has been rejected. By default, a generic message is emitted, indicating that rejection for the sender or recipient has been programmed. You cannot use this qualifier with either the /HEADER qualifier or the /ACCEPT qualifier.

/REGEX

Specifies that the sender/recipient pattern, or header text pattern, uses regular expressions instead of VMS-style wildcards. For header matching, specify /REGEX after the /HEADER qualifier.

ATTACH

Transfers control to another process in the current process tree.

FORMAT

ATTACH *[process-name]*

Command Qualifiers	Defaults
/IDENTIFICATION=process-id	
/PARENT	

PARAMETERS

process-name

Name of the process to which the terminal should be attached. The process must be in the current process tree. This parameter is omitted if one of the qualifiers is specified.

DESCRIPTION

This command is similar to the DCL ATTACH command. It is used in interactive jobs to attach the terminal to another process in the current process tree for the job.

QUALIFIERS

/IDENTIFICATION=process-id

Specifies the process by its process identification, a hexadecimal number.

/PARENT

Specifies that the terminal should be attached to the current subprocess's immediate parent in the process tree.

CHECK

Checks to see if an address would be blocked by the rejection rules.

FORMAT

CHECK *sender-address [recipient-address]*

Command Qualifiers	Defaults
/ADDRESS=ip-address	

PARAMETERS

sender-address

An e-mail address as it would appear on an SMTP MAIL FROM: command, representing the sender of the message.

recipient-address

An e-mail address as it would appear on an SMTP RCPT TO: command, representing the message recipient. If not specified, only the sender address will be checked.

DESCRIPTION

This command simulates the check performed by the SMTP server on a sender address, optionally combined with a recipient address and the numeric IP address of a host.

CHECK will signal its success or failure in finding a matching rule. If successful, the patterns in the rule are displayed in the message.

QUALIFIERS

/ADDRESS=ip-address

Specifies the IP address to be used in the rejection check. This should be specified in dotted-decimal format.

DEFINE/KEY

Defines an equivalence string and a set of attributes with a key on the terminal keyboard.

FORMAT

DEFINE/KEY *key-name equivalence-string*

Command Qualifiers	Defaults
/ECHO	
/ERASE	
/IF_STATE	
/LOCK_STATE	
/LOG	
/SET_STATE	
/TERMINATE	

DESCRIPTION

See the DCL help entry for DEFINE/KEY for more information on this command.

You can have a set of keys defined automatically for use with REJMAN by placing DEFINE/KEY commands in the file SYS$LOGIN:MX_MCP_KEYDEFS.INI. Note that this is the same file that is used with the MCP program.

DELETE EXCLUSION

Removes an exclusion from heuristic junk mail filters.

FORMAT

DELETE EXCLUSION *sender-pat*

Command Qualifiers	Defaults
/HEURISTIC=heuristic-name	

PARAMETERS

sender-pat

E-mail address or wildcard pattern to match the sender of the message, as it would appear in either the RFC822 From header or the SMTP MAIL FROM command.

DESCRIPTION

This command removes an entry from either the global exclusion list, or from the exclusion list for a specific heuristic filter. If the sender (from either the RFC822 From header, or the SMTP MAIL FROM return path) of a message matches this address or pattern, the message is excluded from the specified heuristic filter check (or all checks if the /HEURISTIC qualifier is omitted).

QUALIFIERS

/HEURISTIC=heuristic-name

Specifies the heuristic filter to which this exclusion applies. If not specified, the matching entry is removed from the the global exclusion list for all heuristic filter checks.

DELETE REJECTION

Removes a rejection rule from the database.

FORMAT

DELETE REJECTION *sender-pat [rcpt-pat]*

Command Qualifiers	Defaults
/ADDRESS=ip-addr-pat	
/HEADER	
/IDENTIFIER=rule-id	

DESCRIPTION

This command deletes a rule from the rejection database. Deletion occurs when the specified parameters and qualifiers match the values used when the rule was added.

QUALIFIERS

/ADDRESS=ip-addr-pat

Specifies either an IP address (in dotted-decimal form) or a wildcard pattern for matching an IP address. This qualifier is not allowed when /HEADER or /IDENTIFIER is specified.

/HEADER

Modifies the type of selected rejection rule to be based on RFC822 headers, rather than on SMTP envelope addresses.

/IDENTIFIER=rule-id

Selects the rule to be deleted based on its unique identifier, an integer value assigned when the rule is added to the rejection database.

DISABLE HEURISTIC

Disables junk mail heuristic filters in the SMTP server.

FORMAT

DISABLE HEURISTIC *[heuristic-name...]*

Command Qualifiers	Defaults
/ALL	

PARAMETERS

heuristic-name

A comma-separated list of one or more keywords that identifies the heuristic filters to be disabled. See ENABLE HEURISTIC for the list of heuristic filter names.

DESCRIPTION

This command disables one or more of the heuristic junk mail filters.

QUALIFIERS

/ALL

Disables all heuristic filters. No parameters are permitted when this qualifier is specified.

ENABLE HEURISTIC

Enables junk mail heuristic filters in the SMTP server.

FORMAT

ENABLE HEURISTIC *[heuristic-name...]*

Command Qualifiers	Defaults
/ALL	
/CONFIDENCE_LEVEL={DEFAULT|clevel}	

PARAMETERS

heuristic-name

A comma-separated list of one or more of the following keywords which identify the heuristics to be enabled:

- FROM_TO_SENDER_SAME
- INVALID_AOL_ADDRESS
- INVALID_FROM
- INVALID_HOTMAIL_ADDRESS
- INVALID_TO
- MSGID_HAS_FROM
- MSGID_HAS_TO
- NULL_FROM
- NULL_MSGID
- NULL_TO
- NUMERIC_ADDRESS
- PRECEDENCE_BULK
- RECEIVED_AFTER_FROM
- RECEIVED_ALL_ZEROS
- UIDL_AUTH_SENDER
- X_UIDL

DESCRIPTION

This command enables one or more of the heuristic junk mail filters.

QUALIFIERS

/ALL

Enables all heuristic filters. No parameters or other qualifiers are permitted when this qualifier is specified.

/CONFIDENCE_LEVEL={DEFAULT|clevel}

Specifies the confidence level to be associated with a match against this filter. This is a local qualifier that must be specified immediately after a heuristic name.

The confidence level for a filter is a number representing the likelihood that a message matching that filter is junk mail. A low confidence level indicates that the message is more likely to be legitimate; a high confidence level indicates that the message is more likely to be junk mail. Each message passed through the heuristic filters is assigned the highest possible confidence level for each filter it matches, until either all enabled filters are checked or the message is given a confidence level that is greater than the REJECT level set with the SET HEURISTIICS command.

If the DEFAULT keyword is specified, the built-in default confidence level for the heuristic is set. Otherwise, specify a decimal number from 0 through 10 to set a non-default confidence level.

EXIT

Saves changes to the database and exits the program.

FORMAT EXIT

DESCRIPTION

This command saves the changes made during the current REJMAN session and exits the program.

HELP

Displays information about REJMAN commands.

FORMAT

HELP *[topic]*

PARAMETERS

topic

Topic, such as a REJMAN command, about which you want help. If omitted, a list of available commands is displayed.

DESCRIPTION

This command displays descriptions and other information on REJMAN commands.

QUIT

Exits the program without saving changes.

FORMAT QUIT

DESCRIPTION

This command causes the program to exit without saving the changes made during the current session. You will be prompted for confirmation if you have made any changes.

PURGE

Purges old rejection rules from the database.

FORMAT

PURGE

Command Qualifiers	Defaults
/BEFORE=date-time	*See text.*
/[NO]LOG	*/LOG*

DESCRIPTION

The PURGE command deletes rejection rules from the database considered ''old;'' that is, they were last used by the SMTP server to reject messages before a date in the past. By default, rules not used for 30 days are purged from the database; you may override this by specifying the /BEFORE qualifier.

QUALIFIERS

/BEFORE=date-time

By default, the PURGE command deletes records that have not been used for 30 days. You may override this cut-off date with the /BEFORE qualifier.

/LOG

By default, the PURGE command logs a message indicating the number of rules purged from the database, along with the cut-off date when successful. Specify /NOLOG to prevent this message from being issued on a successful PURGE.

SAVE

Writes out the rejection database to a file.

FORMAT

SAVE *[file-spec]*

PARAMETERS

file-spec

Name of the file to which the database should be written. If omitted, the default is MX_DIR:MX_REJECTION_DATABASE.MXCFG, or the file pointed to by the logical name MX_REJECTION_DATABASE.

DESCRIPTION

This command writes the rejection database to a file. To ensure that the MX SMTP server uses the rejection database information, omit the file specification to allow REJMAN to write the information to the default location.

SET HEURISTICS

Modifies global settings for the heuristic junk mail filters.

FORMAT

SET HEURISTICS

Command Qualifiers	Defaults
/CONFIDENCE_LEVEL=[(ACCEPT=number,REJECT=number)]	
/[NO]INCLUDE_REASON	*/NOINCLUDE_REASON*
/REJECT_ACTION={DROP|FORWARD=address}	

DESCRIPTION

This command modifies the global settings for the heuristic junk mail filter in the SMTP server.

QUALIFIERS

/CONFIDENCE_LEVEL

Sets the confidence levels for accepting and rejecting messages as junk mail. Each heuristic filter has a confidence level, representing the likelihood that a message matching that filter is junk mail. This qualifier is used to set the thresholds for accepting a message as legitimate and rejecting a message as junk mail.

This qualifier takes a list of keywords (ACCEPT, REJECT) that assign the threshold values. Each keyword must be specified with a value, which is either a number from 0 to 10 or the keyword DEFAULT. If you specify both the ACCEPT and REJECT keywords, you must enclose the list in parentheses.

The default settings are:

```
/CONFIDENCE_LEVEL=(ACCEPT=0,REJECT=8)
```

A message is considered fully acceptable when its assigned confidence level is *less than or equal to* the ACCEPT threshold; in that case, the message is passed on to recipients with no further action. A message is considered junk mail when its assigned confidence level is *greater than* the REJECT threshold; in that case, the action specified by SET HEURISTICS/REJECT_ACTION is taken.

Messages assigned confidence levels *between* the two thresholds are considered partially acceptable. They are passed on to the recipients with an additional header (X-Junk-Mail-Rating) warning them that the message may be junk mail. A second additional header containing the reason for the warning may also be included based on the SET HEURISTICS/INCLUDE_REASON setting.

/INCLUDE_REASON

For a message that is assigned a confidence level that falls between the ACCEPT and REJECT thresholds, specifies that an additional header (X-Junk-Mail-Reason) be included in the message to explain to the recipient why the message was flagged with a warning indicating that it might be junk mail. The default is /NOINCLUDE_REASON.

/REJECT_ACTION

Specifies the action to take when the confidence level assigned to a message exceeds the REJECT threshold. This qualifier takes one of two keyword values. The DROP keyword specifies that the SMTP server should drop the message by sending an error status back to the sending SMTP system. The FORWARD keyword specifies that the message should be diverted to a different address. You must specify an e-mail address as a value for the FORWARD keyword.

When the FORWARD keyword is specified, the message is silently diverted to the forwarding address, with headers added to the top of the message indicating the reason the message was rejected, the original sending address, and the original recipients for the message. This setting is provided so that the local postmaster or other responsible person can verify that any messages rejected by the heuristic filters are in fact junk mail. In the case that a rejected message is actually legitimate, the system manager can then forward the message on to its intended recipients and modify the heuristic filter configuration to prevent such messages from being rejected in the future.

The default setting is /REJECT_ACTION=DROP.

SHOW

Displays rejection entries and other information.

FORMAT

SHOW *keyword*

Command Qualifiers	Defaults
/COMMAND	
/IDENTIFIER=rule-id	
/OUTPUT=file-spec	

PARAMETERS

keyword

Specifies the information to be shown. Acceptable keywords are ALL, DATABASE_FILE, HEURISTICS, KEY, REJECTIONS, and VERSION. You must specify one of these keywords.

DESCRIPTION

SHOW DATABASE_FILE displays the name of the database file that you are modifying.

SHOW HEURISTICS displays information about the heuristic filters and exclusions.

SHOW KEY displays key definitions. Refer to the DCL SHOW KEY command for further information.

SHOW REJECTIONS displays the entries in the rejection database, inculding their reference counts, the date and time they were entered in the database, and the date and time they were last used by the SMTP server.

SHOW VERSION displays the version of MX running on the system.

SHOW ALL shows all of the information mentioned above.

QUALIFIERS

/COMMAND

Formats the output as REJMAN commands. This can be used with the /OUTPUT qualifier to create a REJMAN command file to re-create the rejection database from scratch.

/IDENTIFIER=rule-id

Used only with SHOW REJECTIONS, this qualifier selects the rejection rule to be displayed by its unique identifier, an integer value. If not specified, all rejection rules are displayed.

/OUTPUT=file-spec

Directs the displayed information to the specified file. By default, output is displayed on the current standard output device (based on the SYS$OUTPUT logical name).

SPAWN

Spawns a subprocess.

FORMAT

SPAWN *[dcl-command]*

PARAMETERS

dcl-command

A DCL command to be executed in the subprocess. If omitted, REJMAN is suspended and the terminal is attached to the subprocess for interactive entry of commands. To return to the process running REJMAN, use the DCL LOGOUT command to end the subprocess.

DESCRIPTION

This command creates a subprocess to execute one or more DCL commands. See the description of the DCL SPAWN command for further information.

Appendices

A Regular Expressions

MX uses ‘‘extended’’ regular expressions as defined in the POSIX 1003.2 standard. This appendix provides a brief description of the syntax of regular expressions and how they work.

Note: MX regular expression matching is case-insensitive and supports only the ASCII character set.

A.1 Specifying Regular Expressions

A regular expression (RE) is one or more non-empty *branches*, separated by vertical bars. A string matching one of the branches is a considered a match for the RE.

A branch consists of one or more *pieces*, concatenated. A string matching the first piece, followed by the second, etc., is considered to match the branch.

A piece is an *atom* possibly followed by one of the elements listed in Table A–1.

Table A–1 Atom Sequencing Terms

Term	Description
*	Matches zero or more occurrences of the atom.
+	Matches one or more occurrences of the atom.
?	Matches zero or one occurrence of the atom.
{*count*}	Matches exactly *count* occurrences of the atom.
{*min*,}	Matches *min* or more occurrences of the atom.
{*min*,*max*}	Matches *min* to *max* occurrences of the atom, inclusive.

An atom consists of an ordinary character, a *bracket expression* (explained below), or one of the special matching sequences show in Table A–2.

Table A–2 Atom Sequences

Atom	Description
( )	Matches the null string.
.	Matches any single character.
^	Matches the null string at the beginning of a line.
$	Matches the null string at the end of a line.
c	Matches the character *c*, even if it is one of the special characters used for one of the other matching sequences.

A bracket expression typically matches exactly one of a set of characters; the matching set is enclosed in square brackets. Within the brackets, you may enumerate the characters of the set individually or use one of the collating element specifiers described in Table A–3.

Table A–3 Collating Elements in Bracket Expressions

Specifier	Description
a-b	Matches the range of characters between *a* and *b*, inclusive.
^	If the first character in a bracket expression, inverts the meaning; that is, the bracket expression matches any of the characters **not** in the set.
[:*class*:]	Matches any character in the named class. See Table A–4 for a description of the supported character classes.
[:<:]	Matches the null string at the beginning of a word.
[:>:]	Matches the null string at the beginning of a word.
[.*chars*.]	Matches the specified string of characters as a single unit.

To include a right bracket in a bracket expression set, place it at the beginning of the set (immediately after the opening left bracket or ‘‘^’’).

MX recognizes the character class names shown in Table A–4. Only ASCII characters are recognized.

Table A–4 Character Classes

Class Name	Description
alnum	Letter or digit.
alpha	Letter only (A-Z, a-z).
blank	Space or tab character.
cntrl	ASCII control characters.
digit	Digits only (0-9).
domain	Any character permitted in the domain-part of an RFC821/RFC822 address.
graph	Any non-blank printable character.
lower	Lower-case letter.
print	Any printable character, including spaces but excluding tab characters.
punct	Any punctuation character.
space	Any whitespace character, including horizontal and vertical tabs, formfeeds, and carriage returns.
upper	Upper-case character.
username	Any character permitted in the local-part of an RFC821/RFC822 address.
xdigit	Hexadecimal digit.

The *domain* and *username* character classes are MX extensions to standard regular expressions, included to make it easier to match e-mail addresses. Note that since MX always performs case-independent matching, the *upper* and *lower* character classes are equivalent to *alpha*.

Enclosing an RE in parentheses creates a *subexpression*; multiple subexpressions may be concatenated to form a larger RE.

A.1.1 Examples

Some examples of REs are shown below. Note that since periods normally match any character in an RE, to match a literal period in, for example, a domain name, you must quote it with a backslash. The examples are shown in quotation marks as they would be specified in a command; the quotation marks are not part of the REs themselves.

```
"<[[:username:]]+@[[:domain:]]+\.mydom\.com"
```

This RE matches any valid address in any subdomain under ‘‘mydom.com’’.

```
"Subject:[[:blank:]]*\$+[[:blank:]]*MAKE MONEY FAST[[:blank:]]*\$+[[:blank:]]*"
```

This RE would match any Subject: header containing one or more dollar signs, the string ‘‘MAKE MONEY FAST’’, followed by another string of one or more dollar signs (with the dollar signs possibly surrounded by blanks). Note that the text will match regardless of case, and the use of the *blank* character class will match either spaces or tabs.

A.1.2 Substitutions

In rewrite rule processing, strings matching subexpressions are tracked and may be substituted into the rewritten address. While there is no fixed limit on the number of subexpressions allowed in an RE, only the first nine are tracked for substitution purposes. To include the matching string in the rewritten address, specify a backslash followed by the digit (1-9) for the 1st through 9th subexpression, respectively.

For example, the following rewrite rule:

```
DEFINE REWRITE_RULE "<([[:username:]]+)@([[:domain:]]+)\.dnet>" "<""\2::\1""@myhost>"
```

could be used to create a fake ‘‘dnet’’ domain that rewrites domain-style addresses into DECnet-style references for delivery at host ‘‘myhost’’. The ‘‘\2’’ reference causes the string matching the second subexpression (the domain string) to be included, and the ‘‘\1’’ reference causes the string matching the first subexpression (the username string) to be included.

Note that substitution string indices are assigned in order from left to right, based on the presence of the opening left parenthesis, regardless of the level of subexpression nesting. For example:

```
DEFINE REWRITE_RULE "<(user_(alpha|beta))@([[:domain:]]+)>" "<\2@\3>"
```

In this rewrite rule, substitution string ‘‘\1’’ would contain the string matching the entire first subexpression (matching ‘‘(user_(alpha|beta))’’). The ‘‘\2’’ substitution string would contain the string matching the second subexpression, which happens to be nested inside the first (matching ‘‘(alpha|beta)’’). Finally, ‘‘\3’’ would contain the domain-name string to the right of the at sign.

Index

A

address-rewriting rules • 3–1
Alias • 3–2

C

character set • 5–3
component names • 11–1

D

Debugging • 10–2
DEFINE PATH • 3–2
delivery path • 3–2
DNSMTP_INFO file • 10–2
Domain/path mapping • MCP–24

F

File server • MCP–10

H

HDR_INFO file • 10–1

L

LOCAL_INFO file • 10–2
logical names
  MX_SMTP_AUTHFAIL_EVENT_CLASS • 8–14
  MX_SMTP_AUTH_IDENTIFIER • 8–15
  MX_SMTP_LOCK_EXCLUSIONS • 4–3
Logical names
  MAIL$PROTOCOL_prefix • 5–2
  MX_DNSMTP_DEBUG • 10–3
  MX_DNSMTP_SERVER_DEBUG • 10–3
Logical names (cont'd)
  MX_ENVELOPE_FROM_HOST • 5–2
  MX_EVENT_OPER_CLASS • 3–3, 4–4
  MX_FLQ_CHECK_WAIT • 6–2
  MX_FLQ_DEBUG • 10–3
  MX_FLQ_DIR • 1–3
  MX_FLQ_MGR_WAKEUP_INTERVAL • 6–2
  MX_FLQ_PURGE_WAIT • 6–2
  MX_FROM_HOST • 5–2
  MX_LOCAL_CHARSET_DEC_MCS • 5–3
  MX_LOCAL_DEBUG • 10–3
  MX_MLF_DEBUG • 10–3
  MX_PROTOCOL_PREFIX • 5–2
  MX_RESTRICT_USAGE • 5–1
  MX_ROUTER_DEBUG • 10–3
  MX_ROUTER_WAKEUP_INTERVAL • 6–2
  MX_SHUTDOWN • 4–3
  MX_SITE_DEBUG • 10–3
  MX_SMTP_DEBUG • 10–3
  MX_SMTP_PORT • 5–4
  MX_SMTP_SERVER_DEBUG • 10–3
  MX_SMTP_SERVER_MAX_OUTSIDE • 5–4
  MX_SMTP_SERVER_THREADS • 5–3

M

Mail exchanger • 4–1
Mailing lists • MCP–15
MAILQUEUE utility • 9–2
MCP commands
  @ • MCP–4
  ADD USER • MCP–5
  ATTACH • MCP–6
  CREATE USER_DATABASE_FILE • MCP–7
  DEFINE/KEY • MCP–8
  DEFINE ALIAS • MCP–9
  DEFINE FILE_SERVER • MCP–10
  DEFINE INSIDE_NETWORK_ADDRESS • MCP–13
  DEFINE LIST • MCP–15
  DEFINE LOCAL_DOMAIN • MCP–23
  DEFINE PATH • 3–2, MCP–24
  DEFINE REWRITE_RULE • MCP–25
  DEFINE SYSTEM_USERS • MCP–27
  EXIT • MCP–28
  HELP • MCP–29

MCP commands (cont'd)
MODIFY • MCP–30
MODIFY USER • MCP–31
QUEUE CANCEL • MCP–32
QUEUE COMPRESS • MCP–33
QUEUE CREATE • MCP–34
QUEUE DUMP • MCP–35
QUEUE EXTEND • MCP–37
QUEUE HOLD • MCP–38
QUEUE PURGE • MCP–39
QUEUE READY • MCP–40
QUEUE SELECT • MCP–41
QUEUE SHOW • MCP–43
QUEUE STATISTICS • MCP–46
QUEUE SYNCHRONIZE • MCP–47
QUIT • MCP–48
REMOVE • MCP–49
REMOVE USER • MCP–50
RESET • MCP–51
REVIEW • MCP–52
SAVE • MCP–53
SET DECNET_SMTP • MCP–54
SET LOCAL • MCP–55
SET MLF • MCP–58
SET ROUTER • MCP–59
SET SITE • MCP–61
SET SMTP • MCP–62
SHOW • MCP–65
SHUTDOWN • MCP–67
SPAWN • MCP–68
STATUS • MCP–69
MLF agent
settings • MCP–58
MLFAKE utility • 9–1
MLF_INFO file • 10–2
MSG_TEXT file • 10–1
MX__STARTUP.COM • 11–1
MX Control Program • MCP–3
MX_DECODE utility • 9–2
MX_MAILSHR • 5–1
MX_START.COM • 11–1
MX_STARTUP.COM • 11–1

N

next hop • 3–2

P

Percent-hack • 3–3
Process names • 10–2

Q

Queue files • 10–1
Queue file types • 10–1
Queue status • 6–3

R

rewrite rules • 3–1, MCP–25

S

Shutting down MX • 4–3
SITE_INFO file • 10–2
SMTP default router • 4–2
SMTP local domains • MCP–23
SMTP_INFO file • 10–2
SRC_INFO file • 10–1
startup command procedures • 11–1
startup components • 11–1

T

Trace logs • 10–2

U

Utilities
MAILQUEUE • 9–2
MLFAKE • 9–1
MX_DECODE • 9–2

V

VMS MAIL
foreign protocol • 5–2
protocol prefix • 5–2
restricting usage • 5–1