



**MAILMAN**  
**NETWORK REFERENCE GUIDE**

Version 8.0

August 2002

Revised September 2006

Department of Veterans Affairs  
VistA Health Systems Design & Development (HSD&D)  
Infrastructure and Security Services (ISS)



# Revision History

## Documentation Revisions

The following table displays the revision history for this manual. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author(s)
June 1994	1.0	Initial creation of the <i>MailMan Network Reference Guide</i> (Version 7.1).	MailMan Development Team Oakland, CA Office of Information Field Office (OIFO): <ul style="list-style-type: none"> <li>• Maintenance Project Manager—Jack Schram</li> <li>• Project Planner—Laura Rowland</li> <li>• Developers—Gary Beuschel &amp; Lee Hirz</li> <li>• Technical Writer—Thom Blom</li> </ul>
07/23/02	2.0	Initial MailMan V. 8.0 software and documentation release. MailMan V. 8.0 was first released as "DNS-Aware MailMan" in a supplemental document released in August 2002. However, the remaining MailMan documentation set was never updated.	Thom Blom and Gary Beuschel Oakland, CA Office of Information Field Office (OIFO)
09/21/06	3.0	MailMan V. 8.0 documentation reformatting/revision.  Reformatted document to follow the latest ISS styles and guidelines. This is the initial complete reformatting of this manual since its original release with MailMan V. 7.1 in June 1994.  As of this date, all content updates have been completed for all released MailMan patches.  Also, reviewed document and edited for the "Data Scrubbing" and the "PDF 508 Compliance" projects.  <b>Data Scrubbing</b> —Changed all patient/user TEST data to conform to HSD&D standards and conventions as indicated below: <ul style="list-style-type: none"> <li>• The first three digits (prefix) of any</li> </ul>	MailMan Development Team Oakland, CA Office of Information Field Office (OIFO): <ul style="list-style-type: none"> <li>• Maintenance Project Manager—Jack Schram</li> <li>• Project Planner—Laura Rowland</li> <li>• Developers—Gary Beuschel</li> <li>• Technical Writer—Thom Blom</li> </ul>

## Revision History

Date	Revision	Description	Author(s)
		<p>Social Security Numbers (SSN) start with "000" or "666."</p> <ul style="list-style-type: none"><li>• Patient or user names are formatted as follows: XMPATIENT,[N] or XMUSER,[N] respectively, where the N is a number written out and incremented with each new entry (e.g., XMPATIENT, ONE, XMPATIENT, TWO, etc.).</li><li>• Other personal demographic-related data (e.g., addresses, phones, IP addresses, etc.) were also changed to be generic.</li></ul> <p><b>PDF 508 Compliance</b>—The final PDF document was recreated and now supports the minimum requirements to be 508 compliant (i.e., accessibility tags, language selection, alternate text for all images/icons, fully functional Web links, successfully passed Adobe Acrobat Quick Check).</p>	

**Table i. Documentation revision history**

## Patch Revisions

For the current patch history related to this software, please refer to the Patch Module on FORUM.

# Contents

Revision History .....	iii
Figures and Tables .....	ix
Orientation .....	xiii
<b>1. Introduction—MailMan Network Reference Guide.....</b>	<b>1-1</b>
Network MailMan Features .....	1-2
Multimedia Mail .....	1-2
Message Delivery Statistics .....	1-2
Network Delivery Script Error (E Command).....	1-2
Receive NETMAIL from Unknown Domain .....	1-2
Allow Site to Reject Network Mail from Particular Senders .....	1-3
Case Sensitive Message Fields from Network Receptions.....	1-3
Faster NETMAIL Transmission Speeds using TCP/IP Channels .....	1-3
Users Can Create their Own Network Mail Name .....	1-3
Forwarding Addresses Automatically Verified .....	1-3
Directory of Remote Recipients (MailLink) Expanded.....	1-4
Answering a Message Internet Style .....	1-4
Size Limit of Network Transmission.....	1-4
Functional Description .....	1-4
Menu Structure.....	1-6
Network Management Menu .....	1-6
<b>2. Network Architecture .....</b>	<b>2-1</b>
Overview .....	2-1
Management Overview.....	2-1
Components of Network MailMan.....	2-1
Setting up Network MailMan—Devices.....	2-2
Notes on the Mini Out Device .....	2-3
Setting up Network MailMan—Protocols.....	2-4
Error Checking Communications Protocols .....	2-4
Test Protocol.....	2-6
Standards Used by Network MailMan .....	2-6

<b>3.</b>	<b>Application Layer.....</b>	<b>3-1</b>
	Overview .....	3-1
	Sending and Receiving Network Messages.....	3-1
	Steps in the Transmission of Network Messages.....	3-2
	Network Senders Rejected.....	3-2
	Sending Network Messages.....	3-3
	Status of Network Messages Immediately After Sending .....	3-3
	Viewing a Network Message as a User .....	3-3
	What is in the Queue?.....	3-5
	Transmission Queues.....	3-6
	Network Scripts.....	3-7
	Playing a Script & Script Transcripts .....	3-7
	Script for the Big-Site Domain .....	3-7
	Query Display After Message Has Been Sent .....	3-9
	Remote Site Message.....	3-10
	Network Script Commands .....	3-10
	OPEN Command .....	3-11
	SEND Command .....	3-11
	LOOK Command.....	3-11
	CALL Command .....	3-14
	WAIT Command .....	3-14
	FLUSH Command .....	3-14
	DIAL Command .....	3-15
	Command.....	3-15
	XECUTE Command.....	3-16
	MAIL Command .....	3-16
	Script Tips .....	3-16
<b>4.</b>	<b>Network Layer.....</b>	<b>4-1</b>
	Overview .....	4-1
	Syntax for Domain Names.....	4-1
	Domain Name Management .....	4-1
	Domain Name Resolution.....	4-2
	Transmission Process.....	4-2
	Transmission Security .....	4-2

Devices .....	4-3
Queues .....	4-3
Script Processing (Playing a Script) .....	4-3
Progress Report Logic .....	4-4
Postmaster.....	4-4
<b>5. Transport Layer .....</b>	<b>5-1</b>
Overview .....	5-1
Communication Protocols .....	5-2
SCP and 1SCP .....	5-2
Block Mode (3BSCP).....	5-2
SWP .....	5-2
Modem Support.....	5-4
<b>6. Network Mail Troubleshooting.....</b>	<b>6-1</b>
Overview .....	6-1
Transmission Scripts in the Domain File.....	6-1
Avoiding Infinite Transmission Loops .....	6-1
Changes to VistA Domains .....	6-1
<b>7. Network Mail and TCP/IP .....</b>	<b>7-1</b>
Overview .....	7-1
How to Route Internet Traffic through FORUM.....	7-1
Routing to Locally Known Ethernet Address.....	7-1
Starting the TCP/IP Poller .....	7-2
<b>8. Network Mail Across a DECNET Channel .....</b>	<b>8-1</b>
Overview .....	8-1
Setting up a DECNET Channel for MailMan.....	8-1
How to Use a DECNET Channel .....	8-1
Instructions for Completing a DECNET Channel.....	8-1
1. Create an Entry for the DECNET Protocol.....	8-1
2. Enter a Transmission Script in the Domain File .....	8-2
3. Create a LOGIN.COM File on Your Local Machine.....	8-2
4. Define the Remote DECNET Node on Your Machine .....	8-3
5. Create an Account on the Remote Host .....	8-4
6. Issue Appropriate Commands .....	8-5

Contents

7. Create the LOGIN.COM File..... 8-5

8. Define the Remote DECNET Node on the Remote Machine ..... 8-6

**9. Sending and Receiving Messages Between UCIs..... 9-1**

    Overview ..... 9-1

**10. HELO Processing (XMROB) ..... 10-1**

    Overview ..... 10-1

Glossary ..... Glossary-1

Appendix A—MailMan and TCP/IP ..... A-1

    Overview ..... A-1

    Using MailMan and TCP/IP..... A-1

        Arriving ..... A-1

        Sending ..... A-2

        TCP/IP Channel..... A-2

        VMS Process Overview..... A-3

    Setting up MailMan and TCP/IP ..... A-3

        M Process Overview..... A-3

        Maintenance..... A-4

        Installation ..... A-4

        Create and Install Executable Image ..... A-6

        XMINET.COM..... A-9

        M/VMS Software ..... A-11

        M Software ..... A-11

        Other Mandatory Records ..... A-11

Appendix B—TCP/IP Poller..... B-1

    Overview ..... B-1

    Setting up More Domains to Use the TCP Channels ..... B-2

Index ..... Index-1



# Figures and Tables

Table i. Documentation revision history .....	iv
Table ii. Documentation symbol descriptions.....	xiii
Figure 1-1. Network Management menu [XMNET] options.....	1-6
Figure 1-2. Queue Management menu [XMNET-QUEUE-MANAGEMENT] options .....	1-7
Figure 1-3. Transmission Management menu [XMNETE-TRANSMISSION-MANAGEMENT] options.....	1-8
Figure 2-1. Sample modem connection settings .....	2-2
Figure 2-2. Sample DOS/MSM MINIOOUT Physical Link device entries .....	2-3
Table 2-1. Sample protocol logic.....	2-4
Figure 3-1. Sending network messages.....	3-3
Figure 3-2. Status of message immediately after sending—Awaiting Transmission .....	3-3
Table 3-1. Mail message components.....	3-4
Figure 3-3. Sample output when viewing a message as a user .....	3-5
Figure 3-4. Sample queue output when using the Queues with Messages to Transmit Report option.....	3-5
Figure 3-5. Sample output when using the Show a Queue option to see what is in the queue.....	3-6
Figure 3-6. Sample transmission queue output.....	3-6
Figure 3-7. Sample output showing active transmissions .....	3-6
Figure 3-8. Sample script (i.e., ANYWHERE.VA.GOV script) .....	3-7
Figure 3-9. Sample transcript.....	3-8
Figure 3-10. Sample output showing that no more messages are queued for BIG-SITE.VA.GOV .....	3-9
Figure 3-11. Sample output showing query display after a message has been sent.....	3-9
Figure 3-12. Sample remote site message.....	3-10
Table 3-2. Network script commands .....	3-11
Table 3-3. OPEN command format .....	3-11
Figure 3-13. Original format of the LOOK command (1 of 2).....	3-12
Figure 3-14. Sample use of the original LOOK command .....	3-12
Figure 3-15. Updated format of the LOOK command.....	3-12
Figure 3-16. Sample use of the updated LOOK command (1 of 4).....	3-13
Figure 3-17. Sample use of the updated LOOK command (2 of 4).....	3-13
Figure 3-18. Sample use of the updated LOOK command (3 of 4).....	3-13
Figure 3-19. Sample use of the updated LOOK command (4 of 4).....	3-13
Figure 3-20. CALL command format .....	3-14

Figure 3-21. WAIT command format (1 of 2) .....3-14

Figure 3-22. WAIT command format (2 of 2) .....3-14

Figure 3-23. DIAL command format .....3-15

Figure 3-24. DIAL command format—XMSTRIP variable .....3-15

Figure 3-25. ERROR command format .....3-15

Figure 3-26. XECUTE command format .....3-16

Figure 3-27. LOOK command format .....3-16

Figure 4-1. Legal sample domain names .....4-1

Figure 4-2. Illegal sample domain names .....4-1

Figure 5-1. Sample script using the SWP protocol .....5-3

Figure 6-1. Sample TRANSMISSION SCRIPT field entries changing the hardware configuration .....6-2

Figure 7-1. COM file that adds routine information for an IP address .....7-2

Figure 7-2. Starting the TCP/IP Poller using ^XMRTCP .....7-2

Figure 8-1. Sample DECNET entry in the COMMUNICATIONS PROTOCOL File (#3.4) .....8-2

Figure 8-2. Sample transmission script .....8-2

Figure 8-3. Sample LOGIN.COM file .....8-3

Figure 8-4. Sample LOGIN.COM file (with substituted values) .....8-3

Figure 8-5. Sample DECNET node .....8-3

Figure 8-6. Sample account information on remote host (with placeholders) .....8-4

Figure 8-7. Sample commands on the remote host .....8-5

Figure 8-8. Sample LOGIN.COM file .....8-5

Figure 8-9. Sample LOGIN.COM file (with substituted values) .....8-6

Figure 8-10. Sample DECNET node on the remote machine .....8-6

Figure 9-1. Sample DOMAIN File (#4.2) entry .....9-1

Figure 10-1. Sample DOMAIN file (#4.2 entry (with substituted data) .....10-1

Figure A-1. Receiver Flowchart .....A-2

Figure A-2. Sender Flowchart (M Routines/Files) .....A-2

Figure A-3. Stopping INET\_SERVERS .....A-4

Figure A-4. Restarting INET\_SERVERS .....A-4

Figure A-5. Sample VMS account .....A-5

Figure A-6. Sample executable image .....A-9

Figure A-7. Sample XMINET.COM file .....A-9

Figure A-8. XMINET.INFO routine .....A-10

Figure A-9. Sample TWG\$TCP:[NETDIST.ETC]SERVERS.DAT file .....A-10

Figure A-10. Scheduling the XMRTCP to run by invoking XMTCPQUE.COM ..... A-11

Figure A-11. Sample COMMUNICATIONS PROTOCOL file (#3.4) entry ..... A-11

Figure A-12. Sample DEVICE file (#3.5) entry ..... A-12

Figure A-13. Sample TRANSMISSION SCRIPT file (#4.6) entry ..... A-12

Figure A-14. Sample DOMAIN file (#4.2) entry ..... A-12

Figure A-15. Sample TERMINAL TYPE file (#3.2) ..... A-12

Figure B-1. Sample DOMAIN file (#4.2) ..... B-2



# Orientation

This *MailMan Network Reference Guide* is intended for use in conjunction with Veterans Health Information Systems and Technology Architecture (VistA) MailMan. It outlines the network details of the VistA MailMan software and gives guidelines on how the software is used within VistA.

The intended audience of this manual is all primary (key) stakeholders. The primary stakeholders include:

- VistA Infrastructure and Security Services (ISS) Development Team.
- Other VistA project development teams and programmers.
- Information Resource Management (IRM) personnel responsible for maintaining MailMan.
- Enterprise VistA Support (EVS).



## How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of MailMan V. 8.0 and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VistA) software products. This manual discusses the use of electronic network communication software and covers network features for sending and receiving transmissions, responding, and transmitting mail. Many user actions are available for completing specific tasks.

There are no special legal requirements involved in the use of MailMan.

This manual uses several methods to highlight different aspects of the material:

- Various symbols are used throughout the documentation to alert the reader to special information. The following table gives a description of each of these symbols:

Symbol	Description
	<b>NOTE/REF:</b> Used to inform the reader of general information including references to additional reading material.
	<b>CAUTION or DISCLAIMER:</b> Used to inform the reader to take special notice of critical information.

**Table ii. Documentation symbol descriptions**

- Descriptive text is presented in a proportional font (as represented by this font).
- Conventions for displaying TEST data in this document are as follows:
  - The first three digits (prefix) of any Social Security Numbers (SSN) will begin with either "000" or "666".
  - Patient and user names will be formatted as follows: [Application Name]PATIENT,[N] and [Application Name]USER,[N] respectively, where "Application Name" is defined in the Approved Application Abbreviations document and "N" represents the first name as a

number spelled out and incremented with each new entry. For example, in Kernel (KRN) test patient and user names would be documented as follows: KRNPATIENT,ONE; KRNPATIENT,TWO; KRNPATIENT,THREE; etc.

- Sample HL7 messages, "snapshots" of computer online displays (i.e., roll-and-scroll screen or character-based screen captures/dialogues) and computer source code, if any, are shown in a *non-proportional* font and enclosed within a box.
  - User's responses to online prompts will be boldface.
  - References to "<Enter>" within these snapshots indicate that the user should press the **Enter** key on the keyboard. Other special keys are represented within < > angle brackets. For example, pressing the **PF1** key can be represented as pressing <PF1>.
  - Author's comments, if any, are displayed in italics or as "callout" boxes.



**NOTE:** Callout boxes refer to labels or descriptions usually enclosed within a box, which point to specific areas of a displayed image.



**NOTE:** Unless otherwise noted, all sample screen captures/dialogue boxes in this manual are derived from using either MailMan's Detailed or Summary Full Screen message readers.

- This manual refers in many places to the M programming language. Under the 1995 American National Standards Institute (ANSI) standard, M is the primary name of the MUMPS programming language, and M will be considered an alternate name. This manual uses the name M.
- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).

## How to Obtain Technical Information Online

Exported file, routine, and global documentation can be generated through the use of Kernel, MailMan, and VA FileMan utilities.



**NOTE:** Methods of obtaining specific technical information online will be indicated where applicable under the appropriate topic.

### Help at Prompts

Vista M Server-based software provides online help and commonly used system default prompts. Users are encouraged to enter question marks at any response prompt. At the end of the help display, you are immediately returned to the point from which you started. This is an easy way to learn about any aspect of the software.

In addition to the "question mark" help, you can use the Help (User/Group Info., etc.) menu option on the main MailMan Menu to access the MailMan Help Frames through the following options:

- New Features in MailMan
- General MailMan Information
- Questions and Answers on MailMan
- Manual for MailMan Users



**REF:** For more information on obtaining MailMan online help, please refer to Chapter 12, "Online Help Information" in the *MailMan User Guide*.

### Obtaining Data Dictionary Listings

Technical information about VistA M Server-based files and the fields in files is stored in data dictionaries (DD). You can use the List File Attributes option on the Data Dictionary Utilities submenu in VA FileMan to print formatted data dictionaries.



**REF:** For details about obtaining data dictionaries and about the formats available, please refer to the "List File Attributes" chapter in the "File Management" topic of the *VA FileMan Advanced User Guide*.

## Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the following:

- VistA computing environment:
  - Kernel—VistA M Server software
  - VA FileMan data structures and terminology—VistA M Server software
- Microsoft Windows environment
- M programming language

This manual provides an overall explanation of MailMan and the changes contained in MailMan V. 8.0. However, no attempt is made to explain how the overall VistA programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web (WWW) and VA Intranet for a general orientation to VistA. For example, go to the Veterans Health Administration (VHA) Office of Information (OI) Health Systems Design & Development (HSD&D) Home Page at the following Intranet Web address:

<http://vista.med.va.gov/>

## Reference Materials

Readers who wish to learn more about MailMan should consult the following:

- *MailMan Release Notes*
- *MailMan Installation Guide*
- *MailMan Getting Started Guide*
- *MailMan Developer's Guide*
- *MailMan User Guide*
- *MailMan Network Reference Guide* (this manual)
- *MailMan Package Security Guide*
- *MailMan Systems Management Guide*
- *MailMan Technical Manual*
- MailMan Home Page at the following Web address:

<http://vista.med.va.gov/mailman/index.asp>

This site contains other information and provides links to additional documentation.

VistA documentation is made available online in Microsoft Word format and in Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following Web address:

<http://www.adobe.com/>



**REF:** For more information on the use of the Adobe Acrobat Reader, please refer to the "Adobe Acrobat Quick Guide" at the following Web address:

<http://vista.med.va.gov/iss/acrobat/index.asp>

VistA documentation can be downloaded from the Health Systems Design and Development (HSD&D) VistA Documentation Library (VDL) Web site:

<http://www.va.gov/vdl/>

VistA documentation and software can also be downloaded from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO           ftp.fo-albany.med.va.gov
- Hines OIFO           ftp.fo-hines.med.va.gov
- Salt Lake City OIFO   ftp.fo-slc.med.va.gov
- Preferred Method     download.vista.med.va.gov

This method transmits the files from the first available FTP server.





**DISCLAIMER:** The appearance of external hyperlink references in this manual does *not* constitute endorsement by the Department of Veterans Affairs (VA) of this Web site or the information, products, or services contained therein. The VA does *not* exercise any editorial control over the information you may find at these locations. Such links are provided and are consistent with the stated purpose of this VA Intranet Service.



# 1. Introduction—MailMan Network Reference Guide

This manual acquaints system managers and users with the utilities, software structure and functionality of the MailMan networking environment.

MailMan is a form of electronic communication among users sharing computing facilities. A communications link can be made with cables, telephone lines, or satellite connections. In this manner, the networking of electronic mail on a large scale is made possible.

MailMan is designed to allow users to send and receive mail from individuals or groups. A message can take the form of a personal letter or a formal bulletin extracting data from VA FileMan. The text of messages is not difficult to edit, and the context can be made confidential in several ways. Surrogates may be appointed to read mail. Mail groups may be set up to allow each member to respond to a message and to view the replies, as in an informal committee meeting. Mail may be sorted, deleted, forwarded, queried, copied, revised, or printed. In addition, MailMan cross-references messages by subject, message number, sender, and date.

Transmitting information over the network is an involved and complicated process. Over the years the physical media over which information is transmitted has changed, but the basic premises of MailMan's use of the standards and integration with the Veterans Health Information Systems and Technology Architecture (VistA) remain constant. Version 8.0 of MailMan is a great improvement over the original release, and it brings to the VistA environment the ability to transmit Multimedia messages (messages with images and other non-textual parts) across our network.

The Department of Veterans Affairs (VA) communications network, MailMan Version 8.0, is used for interpersonal communications and for the transmission of data to the Austin Automated Data Processing (ADP) Center. Endorsement of this specific functionality has been granted by associated review organizations. This manual provides information for using MailMan V. 8.0.

MailMan V. 8.0 includes automatic interfaces with the VA FileMan software through the use of bulletin cross-references and to parts of Kernel. This version has been designed to be installed separately from the rest of the Kernel software.

MailMan introduces the capability to attach Binary Large Objects (BLOBs) to electronic messages so that images, spreadsheets, graphs and other operating system files that are not pure ASCII text may be sent and received locally and across the network. This is known as Multimedia Mail. Multimedia Mail users will require special terminals and other hardware peripherals to fully use all of its functionalities.

Transmission Control Protocol/Internet Protocol (TCP/IP) channels will enable the use of sending/receiving mail between VistA systems and non-VistA systems (local and remote) running MailMan. This gives VistA systems the fastest, most efficient method for delivering mail.

The effect on performance and system capacity should be minimal. The XMD global has also been added for Site User Directories, which will only impact sites that install Remote User Directories.

## **Network MailMan Features**

### **Multimedia Mail**

When using an imaging workstation, a user can view images that are linked to a message on a second screen. This works using standard calls to the Application Program Interface (API) of the imaging product. Images are brought off a file server onto the Local Area Network (LAN) via Ethernet and are displayed. Network transmissions of messages are performed from a platform that has both Network File System (NFS) capabilities and TCP/IP. Images are delivered off the file server that is on the LAN into a directory that is mounted via NFS onto this platform, again by a call to the imaging API. The relationships of Multimedia hardware must be setup correctly.

The network transmission works by sending a message first. During the exchange of message protocol data units, information is exchanged about the BLOB that is going to be sent via File Transmission Protocol (FTP). The sender indicates that there is a BLOB to send. The receiver replies with the IP address to send the message to and the directory to put it in. The sender then constructs a series of commands that will access the FTP software running on that machine, stores it as a file that can be executed in batch mode, and puts an entry in a global to indicate that the file needs to be executed. A background process in (XMRTCP) polls this file every few seconds and, when it sees an entry, makes a request to the imaging API to move the appropriate image into the export directory. When the image is in the export directory, the aforementioned file is executed and the image is FTP'd to the receiver of the message.

### **Message Delivery Statistics**

Network user and transmission statistical data, including queue waits, number of lines, number of messages, and response times are collected and recorded for statistical reporting. This data is stored in the MESSAGE DELIVERY STATS file (#4.2998).

### **Network Delivery Script Error (E Command)**

System managers can set an error message with "E" command in the transmission script. The error message will tell users the kind of problem that occurred and that the transmission was not successful.

### **Receive NETMAIL from Unknown Domain**

In MailMan you can receive mail from any sub-domain of a domain entry in your DOMAIN file (#4.2).

#### **Example**

Previously, if you had AA.VA.GOV in your DOMAIN file (#4.2), you could not receive mail from SUB.AA.VA.GOV unless SUB.AA.VA.GOV was also in your DOMAIN file (#4.2). With MailMan V. 7.1 or higher you can receive mail from SUB.AA.VA.GOV if the entry for SUB.AA.VA.GOV or VA.GOV or GOV is in your DOMAIN file (#4.2).

## **Allow Site to Reject Network Mail from Particular Senders**

System managers can create a list of senders from whom messages will be rejected. This list contains domain names or users of particular domain names. Therefore, it allows sites to reject all messages that are transmitted from a domain, from a person at a domain, or from any domain that is a sub-domain of a particular domain.

## **Case Sensitive Message Fields from Network Receptions**

MailMan V.8.0 is *not* case sensitive to e-mail addressing. It can handle lowercase, uppercase and mixed case (i.e., both lower and uppercases) of e-mail addresses; however, when sending an e-mail message out to a non-MailMan e-mail system (e.g., UNIX) on the Internet, it is the user's responsibility to enter the correct form of the receiver's e-mail address.

## **Faster NETMAIL Transmission Speeds using TCP/IP Channels**

The advantages of using TCP/IP channels are that TCP/IP channels are very fast and do not require error checking at the application layer. The error checking done at the physical layer of the protocol is CRC-16, which is much better than what is currently used at the application layer (usually LPC). TCP/IP channels are also relatively immune to flow control problems. This method of message transmission and reception is therefore faster, freer from error, and more dependable. It is also a major protocol (the Internet protocol).

## **Users Can Create their Own Network Mail Name**

This is a name that the user assigns to his mailbox on the system.

The Mail name is used as a return address when you send a message to someone whose mailbox is in a different mail environment. It could be different machines in the same facility, an entirely separate facility, or simply a different configuration (or UCI) on the same machine.

The valid format for a mail name is one that is acceptable to all mail systems. It is between 7 and 30 characters in length and contains no punctuation (other than periods). It may contain up to two periods, each of which *must* be preceded and succeeded with at least one alpha or numeric character.

## **Forwarding Addresses Automatically Verified**

When forwarding a message, MailMan will automatically verify the forwarding address. You will receive a "NOT FOUND" message if you try to forward to user that is *not* found in the local domain NEW PERSON file (#200). You will receive a "DOMAIN NOT FOUND" message if you try to forward mail to a user at a domain name that is not found in the DOMAIN file (#4.2).

## Directory of Remote Recipients (MailLink) Expanded

The REMOTE USER DIRECTORY file (#4.2997) contains names and addresses of some remote users. When sending mail to a remote user, you do not have to remember their e-mail address.



**REF:** For more information on the Remote User Directory, please refer to the "MailLink Program" chapter in the *MailMan Systems Management Guide*.

Users can send messages to remote users by entering the remote user's location, mail code or remote user's last name at the "Send Mail to:" prompt.

## Answering a Message Internet Style

An "answer" is different from a "reply" in both format and actions taken. An answer is sent only to the sender of the original message and to recipients you explicitly address your answer to. An answer is very much like a new original message. However, in addition to the text that you type in, the answer also adds two more features to the message. First it copies the original message into your answer text. Then it appends your Network Signature to the end of the answer.

You must have entered a network signature in the Edit User Options in order to answer a message.

## Size Limit of Network Transmission

System managers can now limit the maximum number of lines per message to receive by editing Field #8.31 of the MAILMAN SITE PARAMETERS file (#4.3). System managers can also control the maximum number of lines per message for transmission by editing Field #8.3 of the MAILMAN SITE PARAMETERS file (#4.3) to control the number of lines that may be transmitted by an interactive user without programmer access.

## Functional Description

The Network Management menu [XMNET] is the main MailMan network management menu. It contains options that are used for managing the network-side of MailMan. While the local system requires only some initial setup, the network-side *must* be periodically monitored. The options under the XMNET menu for displaying the queues in various fashions are useful for this purpose. The MailMan network sometimes needs to be flushed. Options for polling and queuing are provided for this purpose. Sometimes, non-delivery of network mail requires investigation. The options for testing a network device and playing scripts are useful in these instances.



**REF:** The Network Management menu [XMNET] and its subordinate options are discussed in depth in the "Network Management Menu" topic in this chapter.

The first options that *must* be used when starting Network MailMan are:

1. Christen a Domain [XMCHRIS]
2. Site Parameters [XMSITE]

While the local system requires some setup, the network-side *must* be monitored. The options under the XMNET menu for displaying the queues in various fashions are useful for this purpose. The network sometimes needs to be flushed. The options under the Transmission Management [XMNET-TRANSMISSION-MANAGEMENT] and Queue Management [XMNET-QUEUE-MANAGEMENT] submenus are provided for this purpose. Sometimes non-delivery of Network Mail requires investigation. There are options for testing of the network device and playing of the script that are useful in these instances.

When a message is created, it has a list of recipients, and usually a textual content. The creation of each message, (whether it is created by software or by users) involves the tasks of maintaining mail groups, mailboxes, and domains. Domains are conceptually the same as computer systems. Each VA Medical Center (VAMC) production account may have multiple machines, but only one domain name. Each recipient's address includes a domain name, either implicitly because the recipient is local to a particular production account where the domain is, or explicitly, because the user's mailbox is located on another system.

System managers need to:

- Provide support in updating the DOMAIN file (#4.2) so that users and applications can effectively communicate across the network.
- Arrange the hardware components for network transmissions.
- Establish physical links that bring systems together temporarily so that messages can be exchanged.
- Understand error checking and how the communication protocols can be used effectively under good and bad conditions.

The transport protocols that are used by MailMan are Simple Mail Transport Protocol (SMTP) and File Transmission Protocol (FTP). SMTP is used by millions of messaging systems that are based on every kind of hardware and operating system, and is a standard for message transmission on the Internet. FTP and SMTP are two of the basic protocols in the suite of protocols known as Transmission Control Protocol/Internet Protocol (TCP/IP). The VA now communicates with many Internet systems through a gateway system provided by a helpful supporter using MailMan SMTP capability.

## Menu Structure

<b>Network Management Menu</b>	<b>[XMNET]</b>
--------------------------------	----------------

The Network Management menu [XMNET] is the main MailMan network management menu. It contains all of the options that are used for managing the network side of MailMan. This menu allows the user to control the network editing, playing, and listing of scripts; managing domains; christening subordinate domains; and controlling pollers. Queuing, queue displays, polling, and testing functions may also be accessed through the use of these options. This menu contains the following options:

```

Select Network Management Option:

    Christen a domain                                [ XMCHRIS ]
    Compare Domains in System Against Released List  [ XMQDOMAINS ]
    Network Help                                    [ XMNETHELP ]
    Queue Management ...                            [ XMNET-QUEUE-MANAGEMENT ]
    Site Parameters                                  [ XMSITE ]
    Transmission Management ...                     [ XMNET-TRANSMISSION-MANAGEMENT ]
  
```

**Figure 1-1. Network Management menu [XMNET] options**

<b>Christen a domain</b>	<b>[XMCHRIS]</b>
--------------------------	------------------

Use the Christen a domain option [XMCHRIS] to christen a domain. This is required before that domain can send or receive networked mail. The subordinate domain must first be initialized with the Initialize Site option at the local domain. Then this option must be selected in order to establish the network relationship.

<b>Compare Domains in System Against Released List</b>	<b>[XMQDOMAINS]</b>
--------------------------------------------------------	---------------------

The Compare Domains in System Against Released List option [XMQDOMAINS] lists domains released with Kernel. It is in a table in routine XMYPOST that is run against the DOMAIN file (#4.2) either with this option or as part of the XMINITs. In either case a report is generated with all the known domains at the time of the release (the Domain file is constantly updated). Those domains not on file at the site are listed and flagged. A summary of the report is also given.

<b>Network Help</b>	<b>[XMNETHELP]</b>
---------------------	--------------------

The Network Help option [XMNETHELP] will display all network help frames.

<b>Queue Management</b>	<b>[XMNET-QUEUE-MANAGEMENT]</b>
-------------------------	---------------------------------

The Queue Management menu [XMNET-QUEUE-MANAGEMENT] contains the following queue management options:



```

Select Queue Management Option:

    Actively Transmitting/Receiving Queues Report           [XMQACTIVE]
    Display Active & Inactive Message Queues               [XMQDISP]
    **> Locked with XUPROGMODE
    Historical Queue Data/Stats Report                      [XMQHIST]
    List a transcript                                       [XMLIST]
    Queues with Messages to Transmit Report                [XMQUEUED]
    Show a queue                                           [XMQSHOW]
    Transmit a Single Queue                                [XMSTARTQUE]
    Transmit All Queues [XMSTARTQUE-ALL]

```

**Figure 1-2. Queue Management menu [XMNET-QUEUE-MANAGEMENT] options**

<b>Actively Transmitting/Receiving Queues Report</b>	<b>[XMQACTIVE]</b>
------------------------------------------------------	--------------------

The Actively Transmitting/Receiving Queues Report option [XMQACTIVE] option produces a report that shows only those queues that are currently (right now) transmitting or receiving messages. It will state the message numbers and the approximate line number in progress and the characters per second.

<b>Display message queues</b>	<b>[XMQDISP]</b>
-------------------------------	------------------

The Display message queues option [XMQDISP] is used by the Postmaster to display the message queues of network sites. This option is locked with the XUPROGMODE security key.

<b>Historical Queue Data/Stats Report</b>	<b>[XMHIST]</b>
-------------------------------------------	-----------------

Use the Historical Queue Data/Stats Report option [XMHIST] to get a report on historical queue data and statistics.

<b>List a transcript</b>	<b>[XMLIST]</b>
--------------------------	-----------------

Use the List a transcript option [XMLIST] to list the transcript for a previously played script.

<b>Queues with Messages to Transmit Report</b>	<b>[XMQUEUED]</b>
------------------------------------------------	-------------------

The Queues with Messages to Transmit Report option [XMQUEUED] produces a report of all those queues that have messages to go out. The list does not say whether there is or is not a task scheduled to send them. Requeuing them is okay even if there is already a task to go out.

**Show a queue****[XMQSHOW]**

The Show a queue option [XMQSHOW] shows all of the messages waiting transmission for a given queue.

**Transmit a Single Queue****[XMSTARTQUE]**

The Transmit a Single Queue option [XMSTARTQUE] is used if messages are queued in the Postmaster's box for a given domain. That domain can begin transmitting those messages by invoking this option. Messages may be sitting in a queue for a variety of reasons (transmission failure, waiting for poller to run, etc.) This option will attempt to send the messages by creating a TaskMan job that will deliver the messages for the domain requested.

**Transmit All Queues****[XMSTARTQUE-ALL]**

Use the Transmit All Queues option [XMSTARTQUE-ALL] if messages are queued in the Postmaster's baskets for a number of domains. TaskMan jobs that will transmit them over the network can be created for all of them by using this option. A domain's queue will not be requeued if it is already set up as a scheduled TaskMan job, nor will it be requeued if it is empty or actively in the process of transmitting.

**Site Parameters****[XMSITE]**

The Site Parameters option [XMSITE] allows the network manager to control local site parameters, names, time zone, etc.

**Transmission Management****[XMNETE-TRANSMISSION-MANAGEMENT]**

The Transmission Management [XMNETE-TRANSMISSION-MANAGEMENT] contains the following transmission management options:

```
Select Transmission Management Option Option:
      Edit a script                               [ XMSCRIPTEEDIT ]
      Play a script                               [ XMSCRIPTPLAY ]
      Receive Messages from Another UCI          [ XMR-UCI-RCV ]
      Send Messages to Another UCI              [ XMR-UCI-SEND ]
      Sequential Media Message Reception         [ XMR-SEQ-RECEIVE ]
      Sequential Media Queue Transmission        [ XMS-SEQ-TRANSMIT ]
      Subroutine editor                          [ XMSUBEDIT ]
      Toggle a script out of service            [ XMSCRIPTOUT ]
      Validation Number Edit                    [ XMEDIT-DOMAIN-VALIDATION# ]
```

**Figure 1-3. Transmission Management menu [XMNETE-TRANSMISSION-MANAGEMENT] options**

**Edit a script****[XMSCRIPTEdit]**

The Edit a script option [XMSCRIPTEdit] allows the creation of a script for a given domain.

**Play a script****[XMSCRIPPlay]**

The Play a script option [XMSCRIPPlay] interprets the chosen script.

**Receive Messages from Another UCI****[XMR-UCI-RCV]**

The Receive Messages from Another UCI option [XMR-UCI-RCV] is the corollary to another called Send Messages to Another UCI. The SEND side loads messages into the %ZISL global. The RECEIVE side reads messages from the %ZISL global.

To use:

1. Log onto the UCI you wish to transmit message from.
2. Use the SEND option. It will load all the messages for a particular domain into ^%ZISL.
3. Log onto the UCI you wish to receive the messages into.
4. Use this option. It will read through the %ZISL global looking for a message to receive (don't worry, it only checks the first line of each potential message/task) and creates a message for them.

Requirements:

1. The receive side needs to have been set up as a domain.
2. You must execute the receive option before the %ZISL global gets cleaned out.

**Send Messages to Another UCI****[XMR-UCI-SEND]**

The Send Messages to Another UCI option [XMR-UCI-SEND] is the corollary to one called Receive Messages from Another UCI [XMR-UCI-RCV].



**REF:** For a complete description of both options and their usage, please refer to the previous (Receive Messages from Another UCI" topic in this chapter.

**Sequential Media Message Reception****[XMR-SEQ-RECEIVE]**

The Sequential Media Message Reception option [XMR-SEQ-RECEIVE] is the converse procedure to XMS-SEQ-TRANSMIT. Please refer to the description of XMS-SEQ-TRANSMIT for more information.

**Sequential Media Queue Transmission****[XMS-SEQ-TRASMIT]**

The Sequential Media Queue Transmission option [XMS-SEQ-TRASMIT] allows the recording of a queue of messages onto sequential media. The messages so recorded may be read into another MailMan system. The Sequential Media Message Reception option is the converse process.

This option has been developed specifically for emergency transmission of messages when the Wide Area Network (WAN) is not available. It can also be used for archiving.

For example, if a bulldozer knocked out your T1 line to the WAN in front of your computer room, it could be three weeks until the system is reconnected to the network. You know that a sister installation 10 miles down the road is still online. You must get your payroll information to Austin. The messages are ready to be sent (in the queue to FOC-AUSTIN.VA.GOV). You can mount a tape and use this procedure to "transmit" the queue onto the tape. Then, have the tape delivered to your sister site (keeping it away from all magnetic fields—those messages are no longer in the queue). At your sister site the tape is mounted and it reads in the messages. The messages are queued up for FOC-AUSTIN.VA.GOV and delivered as though they were relayed through this site. You could also have sent the tape directly to FOC-AUSTIN.VA.GOV where they could have been received and processed.

**Subroutine editor****[XMSUBEDIT]**

The Subroutine editor option [XMSUBEDIT] edits transmission script subroutines.

**Validation Number Edit****[XMEDIT-DOMAIN-VALIDATION#]**

The Validation Number Edit option [XMEDIT-DOMAIN-VALIDATION#] allows sites to edit the validation numbers so that they can be synchronized. The sender and the receiver must edit the numbers concurrently. At S1.VA.GOV they edit S2.VA.GOV's validation number. At S2.VA.GOV they edit S1.VA.GOV's validation number. The same number is used for input at both sites.

## 2. Network Architecture

### Overview

This section describes the design, components, set-up, and maintenance of Network MailMan. Read this section in its entirety to become familiar with the architecture of Network MailMan.

### Management Overview

Network MailMan provides communication services from one computer system (domain) to another. Domains are the center of network activity. In Network MailMan, the DOMAIN file (#4.2), the REMOTE USER DIRECTORY file (#4.2997)) and the Domain Name Server provide the connection between the user and the communication services. It is very important that those who manage Network MailMan understand the various parameters associated with a domain. It is also important that these people understand the network and the various means of providing connections between different computer systems. This manual provides some basic information.

Sources of information on the different layers of Network Architecture are Internet resource material and the Requests For Comment (RFCs). The latter form the basis for the standards used on the Internet. The reader may also want to become familiar with asynchronous and synchronous communications and the current Wide Area Network (WAN) in use by their organization. Network MailMan is part of the public domain and is used by many organizations, including universities, private hospitals, Department of Defense (DoD) hospitals, the Public Health Service (PHS) and the Indian Health Service (IHS).

### Components of Network MailMan

Network MailMan provides services using the following components:

- Name Server
- Script Processor
- Domain Name Server
- Files:

COMMUNICATIONS PROTOCOL (#3.4)	MAILBOX (#3.7)	MESSAGE STATISTICS (#4.2999)
DEVICE (#3.5)	MAILMAN SITE PARAMETERS (#4.3)	NETWORK SENDERS REJECTED (#4.501)
DOMAIN (#4.2)	MAILMAN TIME ZONE (#4.4)	TERMINAL TYPE (#3.2)
INTER-UCI TRANSFER (#4.281)	MESSAGE DELIVERY STATS file (#4.2998)	TRANSMISSION SCRIPT (#4.6)

## Setting up Network MailMan—Devices

The devices that are used for sending and receiving Network Mail are generally released with the Kernel software. There are some things to look out for when setting up a device:

- First, make sure that the device accepts all characters coming into it as they are. TAB characters should not be interpreted or translated into strings of spaces.
- Next, make sure that the device has a large buffer; the larger the buffer, the less likely it is to have problems with network transmissions.
- Be sure to handle DTR and other issues correctly; not properly closing a device can lead to problems with tasks thrashing against devices that are not yet ready for them.

There are two parts to each device. First there is an entry in the DEVICE file (#3.5). Then there is an associated entry in the TERMINAL TYPE file (#3.2). The example given below is used for modem connection:

```

NAME: M-HAYESTXA2                                $I: _TXA2:
ASK DEVICE: YES                                  MODEM: M-HAYESVAX
TASKMAN PRINT A HEADER PAGE: NO                 VOLUME SET(CPU): MAL
LOCATION OF TERMINAL: MODEM LINE TXA2           MARGIN WIDTH: 255
FORM FEED: #                                       PAGE LENGTH: 66
BACK SPACE: $(8)                                   SUBTYPE: M-HAYESVAX
TYPE: TERMINAL                                     TYPE-AHEAD: ALLOWED

NAME: M-HAYESVAX                                RIGHT MARGIN: 255
FORM FEED: #                                       PAGE LENGTH: 66
BACK SPACE: $(8)
CLOSE EXECUTE: D HANG^XMMHAYES S IO("C")=1,ZTNONEXT=1,%ZTIO=" "
DESCRIPTION: HAYES MODEM                            MODEM DIAL LOGIC: D DIAL^XMMHAYES
MODEM HANG UP LOGIC: D HANG^XMMHAYES MODEM STATUS LOGIC: D STAT^XMMHAYES

```

Figure 2-1. Sample modem connection settings

### Asynchronous Communications Links

Network MailMan can be used to communicate with computers that are connected via asynchronous media. Leased lines with modems, statistical multiplexors, or line drivers and Simple Twisted Pair (STP) connections are among the choices available. Auto-dial modems may be used, provided that appropriate modem dial and hang-up logic similar to that shown in the previous example is provided.

### Synchronous Communications Links

Network MailMan can be used across synchronous networks with Transmission Control Protocol/Internet Protocol (TCP/IP channels).

## Notes on the Mini Out Device

It appears that the LatMaster software can be too fast when trying to open ports. TaskMan will open a port to see if it is free before tasking a job to it. If it is free, TaskMan will close the port and immediately have the process requesting a tasked job open the same port. The server is *not* finished cleaning up the port from the CLOSE when the second process tries to open it. It gets caught in a loop of trying to open what it thinks is free, but actually is not. It will then time out or get a data set hang up error. If there is no timeout set, it may appear that tasked jobs take 30-45 minutes (or maybe longer) before they print.

A hang needs to be placed between TaskMan's closing of the port and the second process trying to open the port. A hang of seven seconds is recommended and has been proven to work. Hang times need to be tweaked for each system; some systems may get away with lower hang times than others.

There are two other significant changes:

- Set the OPEN TIME-OUT field in the DEVICE file (#3.5) to seven.
- For active Network Mail ports, enter the following in the CLOSE EXECUTE field of the TERMINAL TYPE file (#3.2):

```
S IO("C")=1,ZTNONEXT=1,%ZTIO=""
```

The CLOSE EXECUTE field drops the current connection so that the device can be used to initiate a new one.



**CAUTION: Not doing this will cause the system to loop indefinitely! The outgoing ports will be re-used without connections being properly dropped. Network Mail tasks will fail repeatedly.**

Additionally, it has been found that setting the LAT ports/queues will also work. Previously, no-queues had been used for host initiated requests. This may be documented in the LAT database. It also contains an added feature to prevent TaskMan from slowing down.

An example of DOS/MSM MINIOUT Physical Link Device follows.

```

OUTPUT FROM WHAT FILE:  DEVICE <Enter>                (168 entries)
NAME:  MINIOUT          $I:  nn
ASK DEVICE:  YES                ASK PARAMETERS:  YES
LOCATION OF TERMINAL:  MINIOUT TO FOC-AUSTIN
MARGIN WIDTH:  225                FORM FEED:  #
PAGE LENGTH:  66                BACK SPACE $C(8)
SUBTYPE:  C-MSM-MINIOUT          TYPE:  TERMINAL
VMS SERVER NODE:  sssss          VMS SERVER PORT:  nn

OUTPUT FROM WHAT FILE:  TERMINAL TYPE                (180 entries)
Select TERMINAL TYPE NAME:  C-MSM-MINIOUT

NAME:  C-MSM-MINIOUT          RIGHT MARGIN:  255
FORM FEED:  #                PAGE LENGTH:  66
BACK SPACE:  $C(8)          OPEN EXECUTE:  X ^%ZOSF("EOFF")
CLOSE EXECUTE:  X ^%ZOSF("EON") S IO("C")=1, ZTNONEXT=1, %ZTO=""

```

**Figure 2-2. Sample DOS/MSM MINIOUT Physical Link device entries**



**NOTE:** X ^%ZOSF("EOFF") is the same as U IO:(:::1) and X ^%XOSF("EON") is the same as U IO:(:::1).

## Setting up Network MailMan—Protocols

The COMMUNICATIONS PROTOCOL file (#3.4) contains information that allows programs to transmit and receive information. The protocols currently supported are Sliding Window Protocol (SWP), 3BSCP (Block Mode), Simple Communications Protocol (SCP), 1SCP (SCP with an operating system checksum), and SERVER. Other protocols may be added by entering the protocol's logic for opening, closing, receiving, and sending.

Protocol Logic	Description
A UNIQUE NAME	For example, SCP.
OPEN LOGIC	Executable M code to be executed when the protocol is first initiated.
CLOSE LOGIC	Executable M code executed when the protocol is ended.
READ LOGIC	Executable M code to read a line of data from the remote device. This returns the data in variable XMRG and sets ER=0 if reception was without error. This code is then loaded into the variable XMREC.
SEND LOGIC	Executable M code to send a line. Data to be sent is in XMSG and ER is set as with a READ.

**Table 2-1. Sample protocol logic**

## Error Checking Communications Protocols

The following protocols are fully checked procedures for reliable transmission over relatively noisy lines (switched modems):

- SCP—The SCP protocols: is reliable, but at the expense of speed and efficiency, since every line is checked line-by-line.
- 1SCP—The 1SCP protocol is more efficient than SCP.
- SWP—The SWP protocols: is more robust than 3BSCP and can deliver messages faster than SCP or 1SCP without failing during periods when communications across the channel are difficult.
- 3BSCP—The 3BSCP protocols: is an improvement that performs block-mode error checking. 3BSCPs: can speed up network transmissions when used, but is very sensitive to problems that may disrupt communications.
- TCP/IP-MAILMAN—The TCP/IP-MAILMAN protocol can be used across TCP/IP channels and is currently the fastest of the protocols, allowing textual information to be sent at speeds 10 to 100 times faster than any of the other protocols.



Speed of transmission and the ability to provide service under less than perfect circumstances are the most important criteria for transmission of messages to remote collection points. Network MailMan started when the VA did not have the sophisticated WAN that they have today.

All the protocols listed above (except TCP/IP-MAILMAN) have error checking functionality at the application level to ensure data integrity. TCP/IP-MAILMAN is better than the others, if a site has the capability to use it, because the TCP/IP channel is synchronous and does the error checking at the level of the physical device.

Network MailMan can be used across modems, satellite links, twisted pair lines, and sophisticated WANs. Care should be taken to choose (on a domain by domain basis) that protocol to use.

The first protocol to be used (SCP) was very slow (i.e., 15-40 characters per second [cps]). The 1SCP protocols:1 was created out of SCP to put the hardest work into an operating system-dependent function and double transmission speeds. 1SCPs:1 is now the slowest protocol a site should consider using unless the site does not have the capability to perform an operating system level Longitudinal Parity Checksum (LPC).

The 3BSCP protocols: transmits large blocks of data rather than a single line, allowing it to use Network Lag-Time. It can take from half-second to three seconds under normal circumstances for data to transverse the WAN. The speed of the 3BSCP protocol under controlled circumstances was over 400 cps.



**NOTE:** The original testing of Network Mail transmission speeds took place between sites located in Washington, Altoona, Austin, and Albany. The Washington site was the originator of most of the messages, but Austin was also a sender. Testing was on a 4800 baud line to and from mini-engines on both sides. The 9600 baud lines were not available at that time. The maximum speed on a 4800 baud line is 480 cps with no error checking overhead.

Because of limitations of the network and of time-loading factors of CPUs receiving transmissions, another protocol was developed that would not work as fast as 3BSCP, but would be less prone to failure under adverse circumstances. This protocol, SWP, will survive many of the problems with communications media and loading factors that 3BSCP cannot. It is not quite as fast as 3BSCPs: under ideal circumstances, running at 250 to 300 cps, but when circumstances are more adverse, it will succeed while 3BSCPs: may have multiple retries.

SWP protocols: also maximizes the use of Network Lag-Time. It checks each line for errors, whereas 3BSCP does an error check for a block of lines. There is more overhead in doing this, but SWP does not need to receive text lines in order. Therefore, if the sender or receiver senses that a line has an error, the line is retransmitted out of order, and received and filed in the correct place in the text on the receiving end. An error using 3BSCP protocols: means retransmission of an entire block. An error under SWP requires retransmission of a single line. Thus, absolute speed is greater using 3BSCP. Errors may cause SWP protocols: to be faster under circumstances where errors are common speed of transmission is dependent on load factors on both the receiving and sending machines. Since READs from the disk are faster than SETs, the receiving site's load is the primary controlling factor in many instances. This is truer for 3BSCP than SWP, as much of the control activity of SWP is oriented toward the sender.

There are multiple protocols to use, depending on circumstances. There should be no need to slow down your transmissions so that the receiving site is accommodated.

### **Current Recommendations:**

TCP/IP-MAILMAN should be used whenever possible. It has the least overhead on the performance of the CPU because all error checking is done at the physical link level of communications. It is the fastest and most dependable. Transmission speeds that approach the band width of the communications media (2000 to 40,000 cps) have been measured.

Use SWP, or use 3BSCP if there is a problem with SWP. If 3BSCP does not work, try either SWP or 1SCP. Use SCP only for "foreign" systems.

### **Test Protocol**

The Test protocol is used for diagnostic purposes. READ comes back to the terminal with a prompt of "R:", allowing the user to enter data. "S:" is displayed when data is sent. This allows a user to test a process by simulating the remote user interface.

### **Standards Used by Network MailMan**

#### **Simple Transport Protocol (SMTP)**

SMTP is a standard that describes how messaging systems using it should communicate to each other. Message Protocol Data Units (MPDUs) are exchanged between systems. The system that is sending the messages initiates each exchange and must send MPDUs that are described in RFCs. The system that is receiving the messages replies with well-defined acknowledgments to the sender's MPDUs. There is a specific order and format for these exchanges. Network MailMan's implementation of SMTP is compatible with all other messaging systems that it has been tested against.

The SMTP was initially used in the Defense Advanced Research Program Agency (DARPA) and described in ARPA RFC 821, a Military Standard approved for use by the DoD.



**REF:** Please refer to the MIL-STD 1781 of the DDN Protocol Handbook - Volume 1.

Now, the Internet is no longer a part of ARPA and new RFCs have supplanted RFC 821, but the original standard has only been enhanced and Network MailMan continues to update itself with these enhancements.

SMTP is the primary transport protocol used by MailMan. SMTP guidelines insure independence from the particular transmission subsystem used for data transfer.

FTP is also used within Network MailMan. FTP is used to transmit Binary Large Objects (BLOBs) via a separate channel. Network MailMan has made proprietary extensions to the SMTP protocol for this purpose, but it does not use these extensions when communicating with non-MailMan systems.



**REF:** Discussion on and an example of setup for VMS users, so that a remote site can send images FTP to your site, is located in the *MailMan Systems Management Guide*.





## 3. Application Layer

### Overview

The Application Layer section of this manual discusses sending, receiving, and viewing messages. It further outlines step-by-step procedures involving message transmission over the network. Information related to playing a script and script transcripts is also found in this section. Below are brief descriptions of all elements of this section.

### Sending and Receiving Network Messages

To name a network recipient:

- You must add a domain name after the recipient's name. The domain name is the network name of the computer system on which the other mail system resides.
- You see the network name of your local system when you log on. MailMan displays "MailMan V. n.n service for Lastname, Firstname at ANYWHERE.VA.GOV" message. ANYWHERE.VA.GOV is replaced by your local site name, as assigned by the network management.
- Network user names must be followed with an at-sign ("@"), followed by their domain name.
- To send a message to XMUSER at domain ANYWHERE-ELSE.VA.GOV, you would respond with, "XMUSER@ANYWHERE-ELSE.VA.GOV", indicating the path the message will take.

MailMan encodes the names of the recipients that it is sending the message to across the network to avoid naming convention problems. On many systems the comma, space and the period have explicit meanings. Therefore names that contain these characters are encoded when sent and decoded when received. For example:

XMUSER,ONE W.@BIGSITE.VA.GOV will be changed to  
XMUSER.ONE\_W+@BIGSITE.VA.GOV

The Remote User Directory allows the user to choose the name and have the system automatically create a network address for the chosen recipient. Other explicit routing mechanisms are being put into place. Users will soon be able to address mail to remote recipients using the !, %, and @ "hacks", as they are amicably called in the Internet community.

This avoids the comma being interpreted as a delimiter between names and the space (forcing a new name context) or the dot being misinterpreted as a separator between labels of the domain. This is done in such a way that the user is shielded from needing to know what is going on in the background. In addition, MailMan has a Remote User Directory that can be used to enter those names commonly used (or automatically loaded with names from another MailMan site using a special option).

The message will be placed in a list, called a queue, along with other messages destined for the same place. Depending on a number of circumstances, the message may be sent immediately or wait for days to be transmitted. Each recipient of the message is tracked independently so users can query the message to find out the status of delivery to any of the recipients. If a message delivery does not succeed because of a

wrong address, the sender of the message will be notified via a separate message of the problem so he can resolve it.

The status of a message immediately after it is sent as Awaiting Transmission, and the path of the message is listed in the PATH field of the query. After it has been sent, the query shows "Data Sent" and "Message ID", indicating the time of transmission completion, the message number, and domain name of the message as it was copied to the remote site.

## Steps in the Transmission of Network Messages

The following illustrates the sequence of steps involved in sending a message over the network. ANYWHERE and BIG-SITE are used as examples.

1. The message is composed and sent with @ Domain Names after the recipient names.
2. A query of the message status shows it awaiting transmission.
3. A display of the queues shows them holding messages.
- 4a. The transmission script to ANYWHERE holds the MailMan instructions for logging onto that system. The script plays, taking about 10 seconds to open the channel, and sends the message.
- or
- 4b. The script to BIG-SITE is more complicated, having to control the M11+ operating system, the VAX, and Kernel. The script to BIG-SITE plays, and after some error recovery, logs onto the VAX, and sends the message via the SMTP protocol.
5. The queues are now empty.
6. The query of the original message shows that it has been sent to the recipients.
7. The remote site shows a new message.
8. The query at the remote site shows the message status and transmission path.

## Network Senders Rejected

The system manager can enter partial or full matches to network addresses into the NETWORK SENDERS REJECTED file (#4.501). Mail from such a sender is rejected. This technique can be used to prevent selected messages coming into a system from a remote site.

## Sending Network Messages

Network messages are composed just as local messages. The network comes into play when recipients are named at remote locations. Mail is then delivered via the network.

```
MailMan n.n service for XMUSER,ONE at ISC-ANY.VA.GOV
You last used MailMan: 14 Jan 90 11:16

Select MailMan Option: SEND MAIL
Subject: TEST MESSAGE
Enter the text of message
  1>This is a test message from the ISC-ANY.VA.GOV domain to
  2>the ANYWHERE.VA.GOV and BIG-SITE.VA.GOV domain.
  3><RET>
Edit Option: <Enter>
Send mail to: XMUSER,ONE// <Enter>
And sent to: XMUSER@ANYWHERE.VA.GOV <Enter>    via ANYWHERE.VA.GOV
And send to: XMUSER@BIG-SITE.VA.GOV <Enter>    via BIG-SITE.VA.GOV
And send to: <Enter>

Select TRANSMIT option: Transmit now// <Enter>
Message sent
```

**Figure 3-1. Sending network messages**

## Status of Network Messages Immediately After Sending

Immediately after sending the message, the status for each recipient is "Awaiting Transmission." Messages are stored in queues for later transmission.

```
Select Message Action: Q

Subj: TEST MESSAGE 14 Jan 90 11:25 2 Lines
From: XMUSER,ONE in 'IN' basket.
Local Message-ID: 1842@ISC-ANY.VA.GOV

XMUSER,ONE          Last read: 14 Jan 90 11:26
XMUSER@BIG-SITE.VA.GOV Status: Awaiting transmission. Path: BIG-SITE.VA.GOV
XMUSER@ANYWHERE.VA.GOV Status: Awaiting transmission. Path:
ANYWHERE.VA.GOV
Select MESSAGE Action: IGNORE (in IN basket)// <Enter>
```

**Figure 3-2. Status of message immediately after sending—Awaiting Transmission**

## Viewing a Network Message as a User

When a user receives a message from a remote domain, it will look just like a local message, except that the FROM field will enclose the sender's name in angle brackets and the domain will follow the sender's name, which is preceded with an at-sign ("@" symbol (i.e., From: <XMUSER@ANYWHERE.VA.GOV>). Also, the time zone of the sending site will be listed in case the remote domain is located in a different time zone. If the user queries a remote message, it will show additional information.

Mail Message Component	Description
Local Message-ID	This is the message identifier of the local version of the message.
Return-path	This is the network name of the sender.
Received	This is a list of the domains through which this message passed to get to you. There is one "received" line per relay.
Message-ID	This is the message identifier of the message at the remote site.
To	<p>This is a list of the recipients of the message as designated by the original sender. There may be more recipients than listed here for the following two reasons:</p> <ol style="list-style-type: none"> <li>1. If there are more than 10 recipients, MailMan will issue the message "too many recipients to list".</li> <li>2. The message may be forwarded locally (or at another recipient's location), which would not show up here.</li> </ol>
In-reply-to	If this message was entered as a reply to other messages, that message identifier is listed here.

**Table 3-1. Mail message components**



**NOTE:** Networked replies are not handled the same as local replies. Replies may show up as separate messages or they may show up as new replies on the reply chain.



```

Subj: TEST MESSAGE 14 Jan 90 10:47 PDT 9 Lines
From: <WILLIAMS@ISC-ANY.VA.GOV> in 'IN' basket.
-----
This is a test message to a number of recipients over the network. The status of
each recipient is displayed with the "Query" option, so that the sender can come
back and see whether the message has been sent yet and, if so, what the message
number at the remote site is. At first, the remote recipients have the status
"awaiting transmission". After they have been sent, the "date sent" and "Message
ID" fields filled in.

Select MESSAGE Action: IGNORE // Q <Enter> (query at the sending site)
Subj: Test message 14 Jan 90 10:47 9 Lines
From: XMUSER,ONE in 'IN' basket.
Local Message-ID: 1838@ISC-ANY.VA.GOV

XMUSER,ONE Last read: 14 Jan 90 11:16
XMUSER@LOMA.VA.GOV Sent: 14 Jan 90 10:47 Message-ID: 36615@ ANYWHERE.VA.GOV
WILLIAMS@BIG-SITE.VA.GOV Sent: 14 Jan 90 10:54 Message-ID: 95626@ BIG-SITE.VA.GOV
WILLIAMS@MAR-AFB.AF.MIL Sent: 14 Jan 90 10:47 Message-ID: 21880@ MAR- AFB.AF.MIL

Select MESSAGE Action: IGNORE // Q <Enter> (query at a remote site)

Subj: TEST MESSAGE 14 Jan 90 10:47 PDT 9 Lines
From: <WILLIAMS@ISC-ANY.VA.GOV> in 'IN' basket.
Message-ID: 36615@ANYWHERE.VA.GOV
Return-path: <WILLIAMS@ISC-ANY.VA.GOV>
Received: from ISC-ANY.VA.GOV by ANYWHERE.VA.GOV; 14 Jan 90 10:50 PDT
Subject: Test message
    Date: 14 Jan 90 10:47 PDT
    Message-ID: <1838@ISC-ANY.VA.GOV>
    To: "XMUSER,ONE"@ISC-ANY,VA.GOV,
        XMUSER@ANYWHERE.VA.GOV,
        XMUSER@BIG-SITE.VA.GOV,
        XMUSER@MAR-AFB.AF.MIL
XMUSER,ONE Last read: 14 Jan 90 11:21
Select MESSAGE Action: IGNORE (in IN basket)// <Enter> Ignored.

```

**Figure 3-3. Sample output when viewing a message as a user**

## What is in the Queue?

Use the Queues with Messages to Transmit Report option.

```

QUEUES WITH MESSAGES TO GO OUT                26 Jun 93 10:49
At FORUM.VA.GOV                               Page: 1

Domain                # Que'd  Physical Link
COLUMBUS.VA.GOV      2      IDCUOUT
PARAGON.COM          1      M-HAYESTXA2
PROVIDENCE.VA.GOV   6      IDCUOUT
S101VAC02.VACO.VA.  33     NULL

Total Domains: 4, Total Messages Queued: 42

```

**Figure 3-4. Sample queue output when using the Queues with Messages to Transmit Report option**

Use the Show a Queue option to see what is in the queue:

```

List of messages for domain PROVIDENCE.VA.GOV
(The Postmaster may READ message headers in the MAIL BASKET with this name.)
*=NEW/+=PRIORITY ##### Subject #####      ### From ###      Responses
                                           Lines      Basket
+1. Koagulab 60s
   [7274361] 06/25/93 @ 15:56           2      XMUSER1,ONE (IRM - Albuq
2. Copy of: BAR CODING
   [7274466] 06/25/93 @ 16:08          388     NOT Selected [PROVIDENCE]
3. 1358 DAILY RECORD
   [7274538] 06/25/93 @ 16:17           6      XMUSER2,TWO (VAMC PROVID
4. INTERIOR DESIGNER
   [7274683] 06/25/93 @ 16:29           6      NOT Selected [PROVIDENCE]
5. COPY OF: DATABASE
   [7275665] 21 JUN 93 11:50           54     XMUSER3,THREE (VAMC LAKE
6. HIV/AIDS INFO
   [7275802] 25 JUN 93 16:15           3      XMUSER4,FOUR (VAMC LEXINGT
                                           NOT Selected [PROVIDENCE]
                                           <"XMUSER5,FIVE_V+"@LIVERMORE.V
                                           NOT Selected [PROVIDENCE]
                                           <XMUSER6,SIX@SANFRANCISCO.VA
                                           NOT Selected [PROVIDENCE]
    
```

**Figure 3-5. Sample output when using the Show a Queue option to see what is in the queue**

A surrogate of the Postmaster can assume the identity of the Postmaster and read the Domain mail basket to see messages in the queue in more detail. Message position in the queue can be prioritized with the "E" command. This basket has the same name as the domain in question and a basket number of 1000+(internal number of the DOMAIN file [#4.2] entry).

## Transmission Queues

The following example shows that there is one message awaiting transmission to BIG-SITE.VA.GOV and two to Kernel.

```

Transmission queue status for ISC-ANY.VA.GOV 14 Jan 90 11:27
--Domain-- Queued Msg line Errors Rate C/S IO
BIG-SITE.VA.GOV      1
KERNEL.SFVDC.VA.GOV  2
ANYWHERE.VA.GOV     1
Totals                4
    
```

**Figure 3-6. Sample transmission queue output**

```

QUEUES ACTIVELY TRANSMITTING MESSAGES                26 Jun 93 10:35
At FORUM.VA.GOV                                     Page: 1
Domain          # Que'd Device/Protocol   Msg #   Line  ZTSK   Errors Rate
C/S
PROVIDENCE.VA.GOV      6          Connecting/Disconnecting
RD4-REGION4.VA.GOV    1  _LTA9801: SWP   7276686  50   1995685  0
121.6
S101VAC02.VACO.VA    . 33  _LTA9802: TCP   8003322 110   1995001  1
900.5
Total Domains: 3, Total Messages Queued: 40
    
```

**Figure 3-7. Sample output showing active transmissions**

## Network Scripts

### Playing a Script & Script Transcripts

The example below (see Figure 3-8) shows the ANYWHERE.VA.GOV script with the connection via direct port-to-port connection to a DSM system. The program tied to that port looks for the word OPEN, then transfers control to the MailMan receiver. The script display is turned off during the process of sending the contents of the message. It is also possible to record transcripts of message transmissions that occur in the background.



**REF:** For more details, please refer to the MAILMAN SITE PARAMETERS file (#4.3).

### Script for the Big-Site Domain

This connection via the IDCU illustrates the "E" commands: that will tell what kind of problem occurred if the transmission was *not* successful. The "E" commands: sets a descriptive phrase, which will be reported if there is an error and the transmission fails.

```

1>E ERROR - Opening Connection
2>O HOST=BIG-SITE,P=SWP
3>S
4>E ERROR - Making Connection across network
5>L:10 USER ID
6>S userid
7>L:10 PASSWORD
8>S password
9>L:20 DESTINATION
10>X W XMHOST,$C(13)
11>L:60 | CONNECTED| |DISCON|
12>E ERROR logging onto remote system
13>S
14>L name: (Logon to VAX)
15>S BIG-SITE
16>S
17>S
18>L CODE: (Logon to KERNEL)
19>S MAIL-BOX SWP
20>L 220
21>E ERROR during message transmission
22>MAIL

```

**Figure 3-8. Sample script (i.e., ANYWHERE.VA.GOV script)**

```

11:11:45 Error message set to 'ERROR - Opening Connection'
11:11:45 Script: BIG-SITE.VA.GOV from ISC-ANY.VA.GOV beginning 01-14-90
11:11:46 Channel opened to BIG-SITE.VA.GOV
11:11:46 Device MINIOUT, Protocol SCP
11:11:46 S:
11:11:47 Error message set to 'ERROR - Making Connection across network'
11:11:47 Look: Timeout=10 Command String = USER ID
11:11:48 R: ~J
11:11:48 R: ~J
11:11:48 R: WELCOME TO THE IDCU
11:11:48 R: ~J
11:11:48 R: ~J
11:11:49 R: ~JUSER ID
11:11:49 S: userid
11:11:49 Look: Timeout=10 Command String = PASSWORD
11:11:51 R: ~JPASSWORD
11:11:52 S: password
11:11:54 Look: Timeout=20 Command String = DESTINATION
11:11:55 R: ~JDESTINATION
11:11:55 Xecuting 'W XMHOST,$C(13)'
11:12:11 Look: Timeout=60, Command String = | CONNECTED| |DISCONN|
11:12:19 R: ~J NNNNNNNNNN Connected
11:12:19 Error message set to 'ERROR logging onto remote system'
11:12:20 S:
11:12:21 Look: Timeout=45 Command String = name:
11:12:34 R: ~J
11:12:34 R: ~J
11:12:40 R: VAA605:977;
11:12:40 R: ~JP 6
11:12:40 R: ~JHOST IS ON LINE
11:12:41 R: ~J~J BIG-SITE
11:12:41 R: Username:
11:12:33 S: BIGSITE
11:12:33 S:
11:12:33 S:
11:12:33 Look: Timeout=45 Command String = CODE:
11:12:34 R: ~J
11:12:34 R: ~J
11:12:35 R: NOTE: Next scheduled P. M. is 1/19/90 at 6:00 A.M.
11:12:35 R:
11:12:35 R: ~JDEVICE TTG2:
11:12:35 R: ~J
11:12:36 R: ~JACCESS CODE:
11:12:36 S MAIL-BOX-SWP
11:12:36 Look: Timeout=45 Command String = 220
11:12:37 R: 220
11:12:38 Beginning sender-SMTP service
11:12:45 R: 220 BIG-SITE.VA.GOV MailMan n.n ready
11:12:55 S: HELO ISC-ANY.VA.GOV
11:12:59 R: 250 OK BIG-SITE.VA.GOV
11:13:12 S: MAIL FROM:<WILLIAMS@ISC-ANY.VA.GOV>
11:13:24 R: 250 OK Message-ID:9560@BIG-SITE.VA.GOV
11:13:30 S: RCPT TO:<WILLIAMS@BIG-SITE.VA.GOV>
11:13:45 R: 250 'RCPT' accepted
11:13:49 S: DATA
11:13:55 R: 354 Enter data
11:13:56 S: Subject:TEST MESSAGE
11:13:59 S: Date:14 Jan 90 11:25 PDT
11:14:02 S: Message-ID:<1842@ISC-ANY.VA.GOV>
11:14:05 S: To: "XMUSER,ONE"@ISC-ANY.VA.GOV,
11:14:09 S: XMUSER@BIG-SITE.VA.GOV,
11:14:13 S: XMUSER@ANYWHERE.VA.GOV
11:14:16 S:
11:14:19 S: This is a test message from the ISC-ANY.VA.GOV domain to the
11:14:19 S: ANYWHERE.VA.GOV and BIG-SITE.VA.GOV domain.
11:14:19 S: .
11:14:33 R: 250 'data' accepted

```

Figure 3-9. Sample transcript

The display of queues shows that no more messages are queued for BIG-SITE.VA.GOV.

```

Transmission queue status for ISC-ANY.VA.GOV 14 Jan 90 11:33
--Domain--      Queued      Msg line  Errors  Rate C/S      IO
WILL.ISC-ANY.VA.GOV
FOC-AUSTIN.VA.GOV
BIG-SITE.VA.GOV
ISC-ALBANY.VA.GOV
KERNEL.SF-ISC.VA.GOV      2
ANYWHERE.VA.GOV
MARCH-AFB.AF.MIL
SF-ISC.VA.GOV
SLC-ISC.VA.GOV
Totals                    2

```

**Figure 3-10. Sample output showing that no more messages are queued for BIG-SITE.VA.GOV**

## Query Display After Message Has Been Sent

The query display of the message shows that the message has been transmitted, and shows the message identifiers at the necessary domains.

```

Subj: TEST MESSAGE 14 Jan 90 11:25 2 Lines
From: WILLIAMS, WILLIAM in 'IN' basket.
Local Message-ID: 1842@ISC-ANY.VA.GOV

XMUSER,ONE                Last read: 14 Jan 90 11:33
XMUSER@BIG-SITE.VA.GOV    Sent: 14 Jan 90 11:30 Message-ID:95650          @BIG-
SITE.VA.GOV
XMUSER@ANYWHERE.VA.GOV  Sent: 14 Jan 90 11:27 Message-ID: 36622@
ANYWHERE.VA.GOV
Select MESSAGE Action: IGNORE (in IN basket)// <Enter>

```

**Figure 3-11. Sample output showing query display after a message has been sent**

## Remote Site Message

The following message will appear at the remote domain.

```
Subj: TEST MESSAGE 14 Jan 90 11:25 PDT 2 Lines
From: <XMUSER1@ISC-ANY.VA.GOV> in 'IN' basket
This is a test message from the ISC-ANY.VA.GOV domain to the
ANYWHERE.VA.GOV and BIG-SITE.VA.GOV domain.

Select MESSAGE Action: IGNORE (in IN basket)// QN

Subj: TEST MESSAGE 14 Jan 90 11:25 PDT 2 Lines
From: <XMUSER1@ISC-ANY.VA.GOV>
Local Message-ID: 36622@ANYWHERE.VA.GOV
Received: from ISC-ANY.VA.GOV by ANYWHERE-.VA.GOV; 14 Jan 90 11:30 PDT

Return-path: ,XMUSER1@ISC-ANY.VA.GOV>
Subject: TEST MESSAGE
Date: 14 Jan 90 11:25 PDT
Message-ID: <1842@ISC-ANY.VA.GOV>
To: "XMUSER1,ONE"@ISC-ANY.VA.GOV,

    XMUSER@BIG-SITE.VA.GOV
    XMUSER@ANYWHERE.VA.GOV
    XMUSER1,ONE
Last read: 14 Jan 90 11:57
Select MESSAGE Action: IGNORE (in IN basket)// <Enter>
```

**Figure 3-12. Sample remote site message**

At the remote domain a QN Query will show the network header information and other details.

## Network Script Commands

Application developers can set up scripts to interface to devices by writing scripts. These script commands can be used in conjunction with calls to the MailMan Script Processor.



**REF:** For more information on the MailMan Script Processor, please refer to "Appendix IX—The MailMan Script Processor" in the *MailMan Developer's Guide*.

This utility is used by MailMan to make connections across the network.

Scripts are used by Network MailMan to establish the linkage between a sender and a receiver. Scripts are managed with the Edit a Script option as text files and are interpreted line by line. The purpose of the script is to identify which outgoing ports are to be used, which queues are to be sent, and which protocols are to be used. It then specifies the sequence of commands to address the network, port switches, modems, gateways, and operating system access codes to get to the desired destination. It establishes communications with the destination domain's SMTP receiver, then drops into the SMTP sender state.

Script commands are stored line by line, beginning with a command name and followed by the command arguments. The script is processed sequentially from the first line. Commands are executed until the last one is executed or one of them fails. If any fail, the transmission is not successful. The commands are:

Command Name	Command Description
OPEN	Open a Device, Queue and Protocol
SEND	Send some text to the Device
LOOK	Look for text from the Device
CALL	Call a Script Subroutine
WAIT	Wait for a specific Number of Seconds
FLUSH	Flush a Line
DIAL	Dial a Modem
ERROR	Set an Error Message
XECUTE	Xecute M Code
MAIL	Initiate Mail Transmission

Table 3-2. Network script commands

## OPEN Command

s:

The format of the OPEN commands: is:

Command Name	Command Description
OPEN	<p>H = [Domain]—Specifies a queue to be sent.</p> <p>D = [Device]—Specifies a device to be used for sending.</p> <p>P = [Protocol]—Specifies a protocol to be used, which must be found in the COMMUNICATIONS PROTOCOL file (#3.4).</p>

Table 3-3. OPEN command format

## SEND Command

The SEND command is used to send text to the device that has been opened. The text will be followed with a single carriage return character (no line feed). A SEND command with no arguments will send a single carriage return. Text may not contain control characters. There is no time-out on the SEND command. A script attempting to send into a device with XOFF pending, for example, may hang indefinitely.



**REF:** For more information, please refer to the "Queue Management" topic in Chapter 1, "Introduction," in this manual.

## LOOK Command

s:

There are two possible formats for the LOOK commands: with MailMan V. 8.0.

## Original LOOK Command

s:

The original LOOK commands: format is as follows:

```
LOOK:[Timeout] Text.
```

Figure 3-13. Original format of the LOOK command (1 of 2)

If [Timeout] is *not* specified, a default of 45 seconds is used.

The command will cause the script processor to examine the input stream until [text] is found on a character-by-character basis using single character reads. As the input comes in, the transcript reads: "Looking for [text]".

The input stream is displayed until [text] is found. Control characters are displayed with their letter value preceded with a "~". LOOK is case sensitive; it looks to match exactly.

```
L:30 CODE
```

Figure 3-14. Sample use of the original LOOK command

Read the character stream for 30 seconds, looking for "CODE".

## Updated LOOK Command

s:

The updated LOOK commands: format is as follows:

```
LOOK:T M{:L} {M{:L}}[TL] {M{:L}}
```

Figure 3-15. Updated format of the LOOK command

Where:

- "T"—Number of seconds until time-out (after which branch to line L2, if the third parameter is defined; else, abort the script).
- "M"—Single word or "|\_a-phrase\_|", with ALL blanks and colons between the vertical bars significant, whose match causes transfer to the appended script line number, "L", if one is found, else to the next line.
- "TL"—Number of the line of the script to jump to on timeout.

There are several format rules:

- A vertical bar ("|") must be the first character after the first space.



**NOTE:** Under old format rules a vertical bar ("|") could *not* be the first character following the first space.



- There must be at least one "M" (there can be more than one). There can be at most one "TL" field.
- The strings identified by "M" must always have a length greater than zero.

### Examples

1. Go to line 3 when " LINE " or " CONNECTED " is encountered. On a time-out just error out.

```
L | LINE | :3 | CONNECTED | :3
```

**Figure 3-16. Sample use of the updated LOOK command (1 of 4)**

2. Look for any occurrence of the four letters "LINE" regardless of the preceding characters and if found, go to line 15 of this script. Look for "CONNECTED" and if found, go to line 18 of this script (note that "DISCONNECTED" contains "CONNECTED"). Go to line 25 of this script on a time-out. If "DISCON" is found, error out.

```
L |LINE| :15 |CONNECTED| :18 25 |DISCON|
```

**Figure 3-17. Sample use of the updated LOOK command (2 of 4)**

3. Same case as 2 except that on a time-out just error out.

```
L |LINE| :15 |CONNECTED| :18 |DISCON|
```

**Figure 3-18. Sample use of the updated LOOK command (3 of 4)**



**NOTE:** "18" is followed by two spaces [time-out is null].

4. If "ON LINE" is encountered, go to line 6. If the string "CONNECTED" is encountered, go to the next line.

```
L |ON LINE| :6 |CONNECTED|
```

**Figure 3-19. Sample use of the updated LOOK command (4 of 4)**

The original syntax for looking for a string used  $\$P(X, ",2,999)$  as the argument, where X is the entire script command. To be backwards compatible, there must be `|s` surrounding all of the strings being searched.

Notice the difference between Examples 1 (Figure 3-16) and 2 (Figure 3-17) with respect to blank spaces. In Example 1, the words "LINE" and "CONNECTED" preceded and followed by a space are sought. In Example 2, the words "ONLINE" or "DISCONNECTED" would also be matched since no blank is required before the word.



**NOTE:** Both the original and updated formats are equally acceptable for performing the LOOK command.

## CALL Command

s:

The CALL commands: format is as follows:

```
C [Transmission Script Name]
```

**Figure 3-20. CALL command format**

[Transmission Script Name] is the name of an entry in the TRANSMISSION SCRIPT (#4.6). When a CALL is made, each command of the entry referenced in the TRANSMISSION SCRIPT (#4.6) is played. This command allows for the creation of scripts that are set up once, then called whenever appropriate.

### Example

C MINIOU calls a subroutine in File #4.6 to connect to a remote CPU.

## WAIT Command

The WAIT command format is as follows:

```
WAIT [seconds]
```

**Figure 3-21. WAIT command format (1 of 2)**

Where [seconds] is the number of seconds that the interpreter should wait before proceeding. The transcript reads:

```
Waiting [num] seconds.
```

**Figure 3-22. WAIT command format (2 of 2)**

## FLUSH Command

The FLUSH command reads all data out of the input buffer to ensure that the data read after that will be clean.

## DIAL Command

s:

The DIAL command format is as follows:

```
DIAL NUM [num]
```

**Figure 3-23. DIAL command format**

[NUM] represents the number to dial.



**NOTE:** The device must have an auto-dial modem properly installed with the appropriate device, sub-type, and modem dialing logic specified.

The numbers are dialed successively until one connects or the list is exhausted. The same number may be specified repeatedly to try successive times.

The code for this command strips all punctuation from the input string. If the XMSTRIP variable contains input characters, these characters will not be stripped from the input.

```
Input = (301),949-6019
XMSTRIP=""
Dialed number=3019496019
XMSTRIP=","
Dialed number=301,9496019
```

**Figure 3-24. DIAL command format—XMSTRIP variable**

### Trying Multiple Phone Numbers when one is Busy

Multiple phone numbers may be passed to the DIAL command such that if the first one is unsuccessful, the next phone number will be dialed, etc. With no other parameters set, the comma is used as the delimiter between each phone number in the string. However, if the variable XMFIELD is set when the dialer is invoked, the dialer will use the following logic to split a dial string into multiple numbers.

If the variable XMFIELD contains a string of characters that is at least one character long, that string will be assumed to be the delimiter between phone numbers. If the variable XMFIELD is not defined or is null, then a comma will be used as the delimiter unless it is one of the characters in the XMSTRIP variable, in which case, a semicolon will be assumed to be the desired delimiter.

## Command

The ERROR command format is as follows:

```
ERROR [MSG]
```

**Figure 3-25. ERROR command format**

This command establishes the text to be displayed if the script fails. This allows the script writer to return a message if the script fails subsequent to this command. The error message is in effect until another error command is issued. In order to cancel an error message, enter an error command with no message.

## XECUTE Command

The XECUTE command format is as follows:

```
X [MUMPS code]
```

Figure 3-26. XECUTE command format

Any M code may be inserted. The transcript will read "executing \_\_\_\_\_".



**CAUTION:** This command should be used with caution, as the full power of the M interpreter is available here.

## MAIL Command

The MAIL command format invokes the SMTP sender, and is generally the last command in the script. The script commands leading up to this command are assumed to have put the remote computer in the SMTP receiver state. Thus, a LOOK 220 command will generally precede the MAIL command.

The MAIL command uses the protocol specified in the OPEN command.

## Script Tips

- The purpose of the script is to reach the remote computer and put it in the receiver state.
- Be careful of timing delays. Waiting a few seconds often makes a script less sensitive to system load.
- When looking for a prompt, use the LOOK commands: to look for the last few characters of the preceding line. For example, to send "ABC" after the "Please Enter Username" prompt of the remote computer, enter:

```
LOOK name  
SEND ABC
```

Figure 3-27. LOOK command format

- Put at least enough text in the LOOK commands: to uniquely identify the sequence you are looking for.





## 4. Network Layer

### Overview

#### Syntax for Domain Names

Domain names must conform to a particular format. Domain names are made up of parts known as labels. Labels are separated from one another by periods. Labels must obey the following rules:

- Must begin with an alpha character.
- Length must be from 1 to 12 characters.
- Contain only alpha characters, numbers, and the dash ("-") character.

Names *cannot* end with a dash ("-"). Domain names are case-insensitive, upper- and lowercase alpha characters are treated the same. Names should be stored in the DOMAIN file (#4.2) in all uppercase format.

```
ABC.DEF.GHI
A123456789.B999999999
TWO-SHOE.SHUFFLE
```

**Figure 4-1. Legal sample domain names**

```
123ABC.DEC (begins with digit)
FORT JONES.VA-GOV (contains blank)
SAN-FRANCISCO-CA.VA.GOV (too long)
```

**Figure 4-2. Illegal sample domain names**

In the above legal examples ABC.DEF.GHI is a third level domain because it consists of three labels. A123456789.B999999999 and TWO-SHOE.SHUFFLE are second level domains because they have only two labels. At VistA locations, all domains contain at least three levels (now they are mostly *site.VA.GOV*). In the future they will contain four levels (*site.MED.VA.GOV*) as the suffix reserved for VistA sites.

### Domain Name Management

The Department of Veterans Affairs (VA) domain names all end in VA.GOV. Domain names used by VistA sites end in either VA.GOV or MED.VA.GOV. Domain names assigned to VistA sites are controlled by POSTMASTER@FORUM.VA.GOV.

After a site has been assigned a name, they have administrative authority to create sub-domains of their domain. The Postmaster at FORUM controls MED.VA.GOV and can assign names to sites, such as BIG-SITE.MED.VA.GOV. At BIG-SITE.MED.VA.GOV, the system manager can assign the name ADMIN.BIG-SITE.VA.GOV to a system that they are associated with. ADMIN.BIG-SITE.VA.GOV

would then be a sub-domain of BIG-SITE.VA.GOV. Thus, naming of many sites can occur with little central coordination.

Don't be confused by the fact that domains that you assign names to are called SUB-DOMAINS. This is an administrative fact, not a judgment on how much power or primacy sub-domains have on the network. If you have any questions, send a message to POSTMASTER@FORUM.VA.GOV or phone FTS 700-427-3700.

## Domain Name Resolution

When a domain name is referenced after the at-sign ("@") in a MailMan address, the following procedures are followed:

1. The full domain name is looked up in the DOMAIN file (#4.2), using standard VA FileMan lookup procedures. For example, LOMA will expand to LOMA-LINDA.VA.GOV.
2. If not found, the left most part of the name is stripped off and the remainder is used. This process is repeated until the name is exhausted or a domain is found. In this manner, TST.LOMA would be found as LOMA-LINDA.VA.GOV, even if the local domain did not have an address for the TST subordinate domain. The message would be routed to the LOMA-LINDA domain, which would then relay it to the TST domain.
3. If none of the above names match, the message "Domain Not Found" is issued and the recipient address is rejected.

Once a domain is found, a queue for transmitting *must* be selected. This process consists of the following:

1. If the domain selected has a relay domain, that domain is selected. This allows the DOMAIN file (#4.2) to force certain domains to certain queues.
2. If not, the domain selected has a transmission script named and is queued for that domain.
3. If not, the message is queued for the domain parent.

## Transmission Process

s:

The transmission process applies to most of the ways that MailMan can be used to transmit messages. It does *not* apply to TCP/IP connectivity except when it discusses how the script processor works.



**REF:** For more information on MailMan and TCP/IP, please refer to Appendix A—MailMan and TCP/IP in this manual.

## Transmission Security

MailMan uses the VALIDATION NUMBER in the DOMAIN file (#4.2) as an added security feature. This feature is used to make communications more secure via Network Mail transmission.





**REF:** For more information on the use of Validation Numbers, please refer to the "MailMan Validation Numbers" chapter in the *MailMan Systems Management Guide*.

## Devices

Upon installation of MailMan, the physical link devices (i.e., the outgoing communications lines) to Wide Area Network (WAN) or modems must be set up. These devices should belong to a Hunt Group. TaskMan will first attempt to find a member of a Hunt Group on the local CPU. If a physical link is not found on the local CPU, the task is moved to a CPU with a physical link. Tasks will be queued to run on the CPU that the device is located on, if the VOLUME SET(CPU) field has a value. Otherwise, it will be assumed that the task can run on any CPU.

## Queues

The word queue is often used to mean a list of things that need to be processed. In the discussion below, it is used to mean a list of messages that need to be sent to a particular domain.

Once a queue has been selected, the message is put in the proper queue, which is a Postmaster's mail basket with the name of the domain. Thus, messages to SITE.MED.VA.GOV are put in the Postmaster's SITE.MED.VA.GOV mail basket. Queues are numbered 1000 and above (1000 + the internal number (pointer) to the related file entry).

MailMan will automatically create a basket if none exists. Currently, messages are placed in baskets according to their internal message number, thereby sending the oldest message first.

If the sender for that queue is inactive, TaskMan is invoked to play the appropriate script in the background. If the "Send Immediate" flag is not on, the message is left in the queue for the remote domain to pick up with the SMTP TURN command, or for the poller to process.



**REF:** For more information on getting a progress report for a domain, please refer to the "Progress Report Logic" topic in Chapter 4, "Network Layer," in this manual.

## Script Processing (Playing a Script)

When a script is invoked in the background, it executes each line successively until completion. If all commands execute successfully, the script quits.

If the script was being processed by a background task and it fails, it reschedules itself for later processing. It will try a number of times. If the script fails repeatedly, a message may be sent to the Postmaster to notify him of problems. The number of times that a script will be tried before such a message is generated is a site-selectable parameter. As messages are sent and received, counters for that domain are incremented.

## Progress Report Logic

As the sender transmits its records, a progress report entry for the domain it is sending is periodically updated. This includes a "\$H" value, a message number, the number of characters, and the number of lines it has transmitted since the script opened.

## Postmaster

The MailMan installation automatically creates a special user (DUZ=.5) called Postmaster, and Postmaster mail baskets. Persons assigned to the management of MailMan and Network mail at a site must be entered in the MAILMAN SITE PARAMETERS file (#4.3) as surrogates of the Postmaster and given Read and Send privileges. A person *must* sign on as himself and then assume the Postmaster's identity for performing Postmaster management tasks.

The Postmaster has mail baskets, and can send and receive messages like any other user. In addition to the IN Basket and Waste Basket, the Arriving Basket is also automatically created for the Postmaster. All mail for a site automatically comes to the Arriving basket from which it is then routed to the individual recipients. The Postmaster may examine the header information of messages, but may not read the text of the message.

Network transmission failures are reported to the Postmaster via bulletins to the Postmaster's IN box. These bulletins should be purged once the problems have been resolved. The Postmaster also needs to check the Arriving basket for purging.

The Postmaster has special Domain baskets (i.e., Birmingham, Dallas, etc.) that are numbered greater than 1000 and are used for the transmission of network mail. Network mail from a site goes to the appropriate Domain basket (queue) prior to transmission. Occasionally the Postmaster may need to stop networked message from going out. This is done by:

1. Deleting the message number from the Postmaster's Domain basket. If a message is deleted from a basket, it is deleted from a queue.
2. Using the RJD Kill off User's Job option to kill transmission of the message to the domain.

The Postmaster also may need to broadcast messages to users. To do this:

1. Send the message to the Postmaster.
2. Assume the identity of the Postmaster.
3. Copy the message into a new message and edit any information that is not appropriate for broadcasting.
4. At the "Send to:" prompt, enter \*.

## 5. Transport Layer

### Overview

The SWP protocol (called Sliding Window Protocol) can speed up Network Mail transmissions when used. It can be over 10 times faster than the 1SCP Protocol and over thirty times faster than the SCP protocol.

The increased speed is due to the decrease in the waiting time that exists across Wide Area Networks (WAN) in the other protocols, measured in seconds. This wait time occurs while the sender is looking for the receiver to return an acknowledgment for the data transmitted. In the SCP and 1SCP protocols, the acknowledgment occurs for each line.

In 3BSCP protocol, lines are blocked into groups of up to 100 lines and sent without waiting for acknowledgment. When the block has been sent, the sender will then pause and wait for the receiver to send an acknowledgment for the entire block of text. If an error has been detected, the block size is reduced and the pause between sending lines is increased until the limit on the amount of reduction has been reached. This process usually results in the sender and receiver properly coordinating and the message transmission being completed.

In order to send or receive transmissions using all protocols except SCP and TCP/IP-MailMan, the sender and receiver must both be capable of producing an LPC checksum.

The SWP protocol is similar to the 3BSCP protocol in the way that it takes advantage of network lag time. The difference lies in how each line is transmitted and therefore how many lines of transmitted data are affected by an error. The SWP sender gets an acknowledgment for each line sent, but does not wait for it. If the sender detects that an acknowledgment has not been received for one line, it indicates that an error has occurred and resends one line. The receiver also detects errors and requests missing lines to be resent. When a re-sent line is received, it has a line number connected with it and the receiver stuffs it into the message in the appropriate position. Therefore, when the connection is less reliable, the SWP protocol resends one line for each error discovered instead of having to resend an entire block of lines as in the 3BSCP protocol.

It is therefore recommended that the SWP protocol be used under all transmission conditions that have many retransmissions associated with them. It is a bit slower than 3BSCP under perfect conditions, but it is very robust under less than perfect conditions.

In the discussion below, ACK stands for positive acknowledgment and NAK stands for negative acknowledgment. An ACK is sent to make the reply, "ok, I understand." A NAK is sent to make the reply, "I didn't get it; something went wrong."

## Communication Protocols

MailMan currently supports four communication protocols:

- SCP—Line-by-line checking
- 1SCP—Line-by-line checking
- 3BSCP—Block-mode checking
- SWP—Sliding Window Protocol

It is recommended that VistA sites run the SWP protocol. Fall back to 3BSCP or 1SCP temporarily, if necessary. The use of SCP is *not* recommended except when trying to communicate with sites that *cannot* calculate a Longitudinal Parity Checksum (LPC).

### SCP and 1SCP

With the SCP and 1 SCP protocols, the sender transmits the message, line number, and checksum, then waits to receive an ACK or NAK. If an ACK appears with the same line number, the message was successfully transmitted and the sender quits. If a NAK appears with the same line number, the message is transmitted again.

The receiver waits for the next line of text. If the line number is the one expected, the receiver reads the checksum. If the checksum is correct, the buffer is flushed, an ACK is transmitted, and the receiver quits. Otherwise the buffer is flushed, a NAK is sent, and the receiver goes back to waiting for the next line of text.

### Block Mode (3BSCP)

Block Mode protocol works very similarly to the SCP and 1SCP protocols. The difference is that the sender sends a line number for each line, but it only sends a checksum for a set of lines. For a NAK from the receiver, the number of lines for a block is reduced and transmission continues, starting from after the previous ACK block.

### SWP

The Sliding Window Protocol (SWP) sends a checksum and line number with each line and expects acknowledgments for each. Under less than perfect conditions, it is the fastest and most dependable protocol. Because it does not wait for ACKs from the receiver, it makes use of network lag time. Because lines may be received out of sequence, a NAK only precipitates the resending of one line (3BSCP *must* resend an entire block).

```

1>O PROT=SWP,HOST=VAX-SITE
2>C MINI
3>S
4>L name:
5>S VAX-SITE
6>S
7>S
8>L CODE:
9>S MAIL-BOX
10>L 220
11>MAIL
12>

Script: VAX-SITE.VA.GOV from ISC-HERE.VA.GOV beginning 01-01-90
Channel opened to VAX-SITE.VA.GOV, MINIOUT Protocol SWP
Calling script 'MINIENGINE'
Waiting 4 seconds
S: A
Looking for 'name:'
R: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
R: J-2462-85-9494
R: Please Log in:
R: ~J
R: ~Juser name:
Xecuting W $C(24,18,8,4),"VA688V: "_XMHOST_" ;password", $C(13)
Looking for ' LINE'
R:
R: ~JP 1
R: ~JHOST IS ON LINE
Returning to script 'VAX-SITE.VA.GOV'.
S:
Looking for 'CODE'
R:
R: Username:
S: VAX-SITE
S:
S:
Looking for 'CODE:'
R: VAX-SITE
R: ~J
R: NOTE: Next scheduled P.M. is 01/11/90 at 6:00 A.M.
R: ~JJDEVICE TTF3:
R: ~JACCESS CODE:
S: MAIL-BOX SWP
Looking for '220'
R: 220
Beginning sender-SMTP service
R: 220 VAX-SITE.VA.GOV MailMan n.n ready
S: HELO ISC-HERE.VA.GOV
R: 250 OK VAX-SITE.VA.GOV
S: MAIL FROM:<WILLIAMS@ISC-HERE.VA.GOV>
R: 250 OK Message-ID:88918@VAX-SITE.VA.GOV
S: RCPT TO:<ZIFF@VAX-SITE.VA.GOV>
R: 250 'RCPT' accepted
S: DATA
R: 354 Enter data
S: Subject:DROPPING INTO PROGRAMMER MODE
S: Date:01 Jan 90 08:48 PST
S: Message-ID:<1502@ISC-HERE.VA.GOV>
S: To: ZIFF@VAX-SITE.VA.GOV
S:
S: User # 1 has dropped into programmer mode on device 101.
S:
R: 250 'data' accepted
S: TURN
R: 502 VAX-SITE.VA.GOV has no messages to export
S:
S:
Script succeeded

```

**Figure 5-1. Sample script using the SWP protocol**

## Modem Support

MailMan supports auto-dial modems that can be controlled by ASCII characters. In order to install an auto-dial modem, the following two steps must be taken:

1. For each port having an auto-dial modem, its entry in the DEVICE file (#3.5) *must* point to the appropriate TERMINAL TYPE file (#3.2) that names the specific modem attached to the port.
2. For each type of modem attachable to Kernel, a specific TERMINAL TYPE file (#3.2) *must* be defined in the file for the modem. Modem Terminal Types begin with **M-**.

Modem Hang Up Logic must contain a valid M expression that accomplishes the actual hang-up operation with the conventions of setting ER=0 (successful) or ER=1 (unsuccessful), and Y containing an explanation for documentation purposes.

Operating system support of modems varies from system to system. Some will automatically drop DTR on the port, which will disconnect the modem if it is configured correctly. The disconnect is sometimes accomplished with the CLOSE of the device, or by a separate routine. MailMan will close the device at the end of a session, but in order to be as portable as possible, MailMan will explicitly hang up the modem using the above logic.

Modem dialer code must control its time-outs carefully. MailMan cannot do so after it passes control to the dialer. All reads must be timed. Remember that the modem may be connected to a long distance telephone line, incurring charges while a READ may be hanging indefinitely.

When writing a modem dialer, it may occasionally be necessary to insert HANGs in your code to keep the M code from overrunning the modem logic. This has to be carefully checked. Sometimes it also helps to reset the modem if possible.

Begin your script with an open statement. In this statement you will name the host and protocol. Name the device only if the PHYSICAL LINK field for the domain has not been filled in.

## 6. Network Mail Troubleshooting

### Overview

There are many reasons why Network Mail can fail to be transmitted. Some problems that occur most frequently are listed below.

### Transmission Scripts in the Domain File

MailMan looks at the Transmission Script field in two places. The most obvious is a script that is interpreted by the script processor. This gives results that are explicitly defined.

### Avoiding Infinite Transmission Loops

This field is also looked at by the name server to determine implicit routing. The implication to the system if this field is not defined, is that the local system does not know how to deliver mail to this domain. There are no directions in the form of a transmission script to tell MailMan what to do. Since this system does not know how to deliver the message, it is assumed that the parent of this system does know. Therefore, if a domain has no data at all in the TRANSMISSION SCRIPT field, MailMan routes messages to the local domain's parent. This often occurs with sub-domains. No protection exists to prevent infinite loops of transmissions that can occur when the parent has no entry for the sub-domain and therefore implicitly assumes that it should route the message to the sub-domain's creator.

### Changes to VistA Domains

Until early 1991, domain names were used only by VistA sites within the VA. In early 1993 the VA reorganized administrative responsibility for domain names by assigning a 3rd level domain to each major sub-organization within the VA. Because of this reorganization, VistA domain names now will end with the ".MED.VA.GOV" suffix.

The heart of the Network Mail system of the Veterans Health Information Systems and Technology Architecture (VistA) is the DOMAIN file (#4.2). Information in this file must be correct for the messages to be delivered properly.

It is important that a guideline exist for reporting and implementing changes to the DOMAIN file (#4.2) to all sites running VistA software. This guideline is intended to save time for all the system managers who must ensure that their electronic mail is delivered and received in a timely manner.

The DOMAIN file (#4.2) in MailMan contains the information that sites use to communicate via Network Mail. In this file, there are domains that are used in a small group of sites and domains that should be known by all VistA sites. This document applies only to those domains whose use is intended for all VistA sites (nationally).

Information that is distributed with the DOMAIN file (#4.2) *must* be kept current. All sites should have timely updates of changes to the DOMAIN file (#4.2) that will affect their ability to communicate. A mail

group has been established on FORUM, called DOMAIN CHANGES. This mail group will be used to report changes to domains or requests for new domains that are distributed nationally (address mail to G.DOMAIN CHANGES). Changes that do not affect domains that must be distributed nationally do not need to be coordinated through the structure established by this guideline. Subordinate domains, or locally created domains such as domains used within a district or region, do not need to conform to this guideline. Creation of and changes to these domains should be handled locally. The local IRM Chief is accountable for their accuracy.

It is the responsibility of each site to maximize the efficiency of network communications with their site. Significant changes to the DOMAIN file (#4.2) that should be coordinated include the following:

- A change to the MailMan Host field in the DOMAIN file (#4.2). The value of this field is used to tell the network which gateway to use. It can narrowly define the connection to a single port or group of ports.
- A change to the hardware configuration that affects the script for the domain. Changing the hardware configuration can mean that the TRANSMISSION SCRIPT field must change. It may be necessary or desirable to change either the text and/or the protocol used. For example, if the configuration changes such that remote mail cannot connect with the following script, it will have to be justified.

```
1>0 H=DOMAINNAME
2>C MINI
3>C KERNEL
```

**Figure 6-1. Sample TRANSMISSION SCRIPT field entries changing the hardware configuration**

- A change to a domain's name (approval to change an existing domain name will rarely be given).
- A request for establishment of a new domain.

The method for accomplishing coordination is to send a message indicating the change(s) to G.DOMAIN CHANGES on FORUM. This should be done at least 14 working days before the change is to be made, for the development team to issue a patch and coordinate the necessary change. Electronic mail services must not be interrupted.

Information that must be provided includes:

- The reason for the change.
- The change(s) that need to be made for the DOMAIN file (#4.2)
- Those applications or individuals concerned/affected.
- The purpose of the domain, if the information concerns a new domain.

It is the responsibility of the development ISC for MailMan to issue patches to the MailMan system for changes to the DOMAIN file (#4.2).



**NOTE:** A patch, XM\*DBA\*42 (with associated automatic installation routines XMYMNEM\*) was issued to insert mnemonics into the MailMan Host fields of the DOMAIN file (#4.2). This patch should be installed. No patches to MailMan Host Numbers will be released in the future, but mnemonics will work regardless of changes to network configurations.







## 7. Network Mail and TCP/IP

### Overview

:

#### How to Route Internet Traffic through FORUM

Automatic forwarding of mail messages from MailMan to an Internet user is now available in this version. Any Internet address that ends with one of the suffixes listed below (almost all) can be used simply by typing in the address to send to (as you would with any VistA domain).

Here are some of the many possible suffixes in the DOMAIN file (#4.2):

ARPA	EDU	O
BITNET	FI	RG
CA	GOV	UK
COM	MIL	UUCP
DE	NET	ZA

The message would then be routed through FORUM and sent through our Internet gateway. From there, MailMan is SMTP compliant. It can communicate directly with other SMTP compliant messaging services. Most messaging services on the Internet are SMTP compliant.

#### Routing to Locally Known Ethernet Address

In order to FTP to our "Locally Known" Ethernet Address, 89.0.0.99, you must have an entry for 89.0.0.99 in your routing table. You can view the table from the VMS "\$" prompt by typing in:

```
netstat -nr
```

This will give you a list of the current routing table.

To add the necessary entry (if it doesn't already exist) type in:

```
route add 89.0.0. 1.2.0.1.0
```

This routes any attempts to reach 89.0.0.99 through 1.2.0.1 with 0 hops.

**Example:**

```

$! route_add.com 6/2/92 DEB --> VAX SITE command procedure to insert into
$! twg$tcp:[netdist.misc]startinet.com to enable routing so that a HOST at
$! a SITE A can reach any HOST at SITE B.
$! Insert towards end of the twg$tcp:[netdist.misc]startinet.com, after call
$! IDCU.COM
$!
$! for SITE A
$! to enable routine, replace l.x.y.z with the appropriate X.25 IP address of
$! the "B" HOST connected to IDCU:
$! replace 200.a.b.0 with the appropriate LAN IP address of the REMOTE SITE B.
$!
$! route add 200.a.b.0 l.x.y.z 1 !to get to REMOTE SITE B
$!
$! for SITE B
$! to enable routing, replace l.m.n.o with the appropriate X.25 IP address of
$! the "A" HOST connected to IDCU:
$! replace 200.c.d.0 with the appropriate LAN IP address of the REMOTE SITE A.
$!
$! route add 200.c.d.0 l.m.n.o !to get to REMOTE SITE A
$!
$! PCs would need to probably just define a "default gateway" or "gateway
$! address as the local VAX that will be doing the routing.

```

**Figure 7-1. COM file that adds routine information for an IP address****Starting the TCP/IP Poller**

:

From a privileged account, start the ^XMRTCP routine as shown below:

```

$!Start _TCP_ Filer
$ run sys$system:loginout/input=tcpdet.com/output=tcpdet.log/detach -
/ast_limint=300/buffer_limit=40960/enqueue_limit=300/file_limit=150 -
/io_buffered=64/io_direct=64/job_table_quota=1024/maximum_working_set=1864 -
/page_file=10240/queue_limit=10/working_set=900/subprocess_limit=30
$! TCPDET.COM - Start the TCP MailMan Poller Process
$ set noon
$ dsm/env=dsmmgr/uci=vah/vol=rou ^XMRTCP
$ exit

```

**Figure 7-2. Starting the TCP/IP Poller using ^XMRTCP**

**CAUTION:** Also change the code at JOB^XMRTCP to the same code as shown in Figure 7-2. Keep a DCL context and you can then spawn out your FTP scripts, as well as a number of other things.

## 8. Network Mail Across a DECNET Channel

### Overview

:

#### Setting up a DECNET Channel for MailMan

The following should provide you with the information necessary to set up a link via DECNET. MailMan V. 3.18 or higher and Kernel V. 6 or higher is a requirement.

On your machine (Local MailMan):

- Set up an entry in the COMMUNICATIONS PROTOCOL file (#3.4) called DECNET.
- Enter Transmission Script in the DOMAIN file (#4.2)
- Create a LOGIN.COM file (for use by the remote host).
- Define the remote DECNET node on your machine.

On the remote host:

- Create an Account on the remote host.
- Create an Object on the remote host.
- Create LOGIN.COM file on the remote host.
- Define your DECNET node on the remote machine.

### How to Use a DECNET Channel

The DECNET task names used in the following can be changed. You will have to use different names under different circumstances. You need to be consistent when setting up each channel. The work you do in Steps 1, 3, and 4 is not repeated for each additional channel you create. The task name used here is "DECNET\_TASK\_NAME".

### Instructions for Completing a DECNET Channel

After completing these steps, you will be ready to test the connection. Feel free to contact the Washington ISC for assistance.

#### 1. Create an Entry for the DECNET Protocol

Create an Entry for the DECNET Protocol in the COMMUNICATIONS PROTOCOL File (#3.4).

```

NAME:          DECNET
SEND:         W XMSG,! D ENS^XMC11
RECEIVE:      R XMRG S ER=0 D ENR^XMC11
OPEN:        S XMLST=$H*86400+$P($H,"",2) I '($D(IO)#10) S (IO,IO(0))=$I
CLOSE:       K XMLINE,XMLCT,XMLST Q
DESCRIPTION:  This protocol is designed exclusively to be used with DECNET TASK
to TASK communications. While using the DECNET channel *READS and #READS may
not be done.

```

**Figure 8-1. Sample DECNET entry in the COMMUNICATIONS PROTOCOL File (#3.4)**

## 2. Enter a Transmission Script in the Domain File

Enter a transmission script in the DOMAIN file (#4.2) for the domain with which you are going to communicate with over this link. Also, change the PHYSICAL LINK field to "\_NLA0:". This is the logical name for the Null device. If you have an entry in the DEVICE file (#3.5) for the Null device, the name of this device can be used. Otherwise, you should create such an entry. It is best to have a device in the DEVICE file (#3.5) with \_NLA0: as the \$I, and to use the name of this device in the PHYSICAL LINK DEVICE field.

The script looks like the one below (see Figure 8-2), where "xxxx" is the domain name you are setting up. It is also a variable name that is ZALLOCATED to avoid multiple jobs to the same domain on the same channel.

### Example:

node = DECNET node name of remote host.

```

1> ZA node:1 E S ER=1
2> O H=xxxx.VA.GOV,PROTOCOL=DECNET
3> X S IO="node""::"TASK=DECENT_TASK_NAME"" O IO
4> X S XMHANG="C IO S IO=0,IOST(0)=0"
5> LOOK 220
6> MAIL
7> ZD node

```

**Figure 8-2. Sample transmission script**

## 3. Create a LOGIN.COM File on Your Local Machine

Create a LOGIN.COM file on your local machine. This is done so that the remote domain CPU can communicate to your machine (domain) using the DECNET transmission protocol/script. If this is not done, the remote site will still be able to communicate with you using the SCP, 1SCP SWP, and 3BSCP protocols and scripts.

On the last line of the file is the code that invokes the M tag^routine (DECNET^XMR). Change lowercase characters to the appropriate values when you create your own.

**Pattern:**

```

$!NETMAILvol_LOGIN.COM
$!Command file for netmail group ??? user over decnet
$ if f$mode() .nes. "NETWORK" then logout
$DSM/VOL=vol/UCI=uci/PACK/TYPE/NOENABLE DECNET^XMR

```

**Figure 8-3. Sample LOGIN.COM file****Example:**

The volume used in this example is AAA and the UCI is VAH:

```

$!NETMAILAAA_LOGIN.COM
$!Command file for netmail group ?? user over DECNET
$ if f$mode() .nes. "NETWORK" then logout
$DSM/VOL=AAA/UCI=VAH/PACK/TYPE/NOENABLE DECNET^XMR

```

**Figure 8-4. Sample LOGIN.COM file (with substituted values)****4. Define the Remote DECNET Node on Your Machine**

Define the remote DECNET node on your machine. This is done through NCP software. It may be accessed and updated as follows from the VMS "\$" prompt (nnnn.nn = DECNET node ID#).

```

$NCP:=$NCP
$ MCR NCP
NCP> DEFINE NODE node ADDRESS nnnn.nn

```

**Figure 8-5. Sample DECNET node**

## 5. Create an Account on the Remote Host

On the remote host, create an account substituting your values where indicated:

```

DISK$whatever=The disk that holds the login directory
?Directory=The directory for logins
DSM? group=The group of the DSM Configuration ex:[50,2]
PASS???=The password for the account
vol=Same as in Step #2
UAF> ADD NETMAILvol/device=DISK$whatever:/dir=[?directory]fillm=60-
/wsdef=500/wsquo=1000/wsext=1700/pgflquo=100000/enqlm=2000/tqelm=50-
/astlm=50/diolm=50/biolm=1000/uci=dsm ? group/lgicmd=NETMAILvol_LOGIN-
/passwork=PASS???:/priv=(group,grpnam,tmpmbx,net,bx)-
/defpriv=(group,grpnam,tmpmbx,netmbx)/bytlm=100000/prclm=10/queprio=4
/NOPWEXPIRED/PWDLIFE=NONE/NOACCESS/NETWORK

Username: NETMAILvol                               Owner: DSM
Account: xxxxxx                                     UIC: same group as DSM
CLI: DCL                                           Tables: DCLTABLES
Default: DISK$whatever:[NETMAIL]
LGICMD: NETMAILvol_LOGIN
Flags:
Primary days: Mon Tue Wed Thu Fri
Secondary days:                                     Sat Sun
Primary 000000000011111111112222 Secondary 000000000011111111112222
Day Hours 012345678901234567890123 Day Hours 012345678901234567890123
Network: ##### Full access #####                ##### Full access #####
Batch: ---- No access ----                      ---- No access ----
Local: ---- No access ----                      ---- No access ----
Dialup: ---- No access ----                    ---- No access ----
Remote: ---- No access ----                    ---- No access ----
Expiration: (none) Pwdminimum: 6 Login Fails: 3
Pwdlifetime: (none) Pwdchange: 23-AUG-1993 11:35:29
Last Login: (none) (interactive), 22-MAY-1993 17:49:30
Maxjobs: 0 Fillm: 6 Bytlm: 100000
Maxacctjobs: 0 Shrfillm: 0 Pbytlm: 0
Maxdetach: 0 BIOlm: 1000 JTquota: 1024
Prclm: 10 DIOLm: 50 WSdef: 500
Prio: 4 ASTlm: 50 WSquo: 1000
Queprio: 4 TQElm: 50 WSextent: 1700
CPU: (none) Enqlm: 2000 Pgflquo: 100000
Authorized Privileges:
GROUP GRPNAM TMPMBX NETMBX
Default Privileges:
GROUP GRPNAM TMPMBX NETMBX
    
```

Figure 8-6. Sample account information on remote host (with placeholders)



## 6. Issue Appropriate Commands

On the remote host, issue the following commands. In this example, the password is DECNET\_TASK\_TO\_TASK\_PASSWORD.

```

$MCR NCP
NCP>DEF OBJECT NETMAILvol NUMBER 0 FILE NETMAILvol_LOGIN-
USER NETMAILvol PASS DECNET_TASK_TO_TASK_PASSWORD
NCR> SET OBJECT NETMAILvol ALL
NCP>EXIT

NCP>SHO OBJ NETMAILvol CHAR

Object Volatile Characteristics as of 29-JUK-1993 18:57:15
Object = NETMAIL vol

Number      = 0
File id     = NETMAILvol_LOGIN.COM
User id     = NETMAILvol
Password    = DECNET_TASK_TO_TASK_PASSWORD

```

**Figure 8-7. Sample commands on the remote host**

Be sure that the password entered here matches the password entered for the account you just created. The network object logs into VMS through the User Authorization file (UAF) and invokes the Command file listed to call DECNET^XMR.

## 7. Create the LOGIN.COM File

Create the following LOGIN.COM file in the default directory of the account you just created. This file serves as the LGICMD for the netmail username and is also the file listed for the NCP Object file.

This is done so that your domain CPU can communicate to the remote domain (machine) using the DECNET transmission protocol/script also. If this is not done, the remote site will not have a way for your task to become a process. Each process must log-in through a Com file when DECNET task-to-task communications are used.

### Pattern:

```

$!NETMAILvol_LOGIN.COM
$!Command file for netmail group ?? user over decnet
$ if F4mode() .nes. "NETWORK" then logout
$DSM/VOL=vol/UCI=uci/PACK/TYPE/NOCENABLE DECNET^XMR

```

**Figure 8-8. Sample LOGIN.COM file**

**Example:**

The volume for this example is BBB and the UCI is VAH.

```
$!NETMAILBBB_LOGIN.COM
$!Command file for netmail group ?? user over decnet
$ if f$mode() .nes. "NETWORK" then logout
$DSM/VOL=BBB/UCI-VAH/PACK/TYPE/NOCENABLE DECNET^XMR
```

**Figure 8-9. Sample LOGIN.COM file (with substituted values)**

## 8. Define the Remote DECNET Node on the Remote Machine

Define the remote DECNET node on the remote machine. This is done through the NCP software. It may be accessed and updated as follows from the VMS "\$" prompt (nnnn.nnn = DECNET node ID#).


```
$ MCR NCP
NCP> DEFINE NODE node ADDRESS nnnn.nnn
```

**Figure 8-10. Sample DECNET node on the remote machine**

# 9. Sending and Receiving Messages Between UCIs

## Overview

The options contained in the Transmission Management menu [XMNET-TRANSMISSION-MANAGEMENT] require you to have UCIs set up, christened, and working as MailMan sites.


 **REF:** For more information on UCI setup, please refer to the *MailMan Technical Manual* and *MailMan Systems Management Guide*.

An additional requirement is that UCIs share a manager UCI (often named "MGR"). This is because the process consists of two steps:

1. Messages in the queue are "dumped" to ^%ZISL.
2. Messages in ^%ZISL are read and moved into the Message file.

To set up the UCIs properly, make sure that the following is correct:

- Both of the UCIs must be christened.

 **REF:** For more information on christening UCIs, please refer to the "Christen a domain" topic in Chapter 1, "Introduction," in this manual.

- The names of both UCIs (their domain names) must be entries in the DOMAIN files (#4.2) in both UCIs.

When this is true, send, then receive a test message. Query the message and make sure that the recipient is marked as you would expect it to be marked. Do this in both directions, once from UCI AAA to UCI BBB and once from UCI BBB to UCI AAA.

Each DOMAIN file (#4.2) entry made for this purpose should have at least the following information. Make sure there is a transmission script with the text. If there is no transmission script text, MailMan will attempt to route messages to your domain's parent. Since it will not really be used to transmit messages, it needs only to have any sequence of characters on a single line, such as, "X Q", as shown below:

```
NAME:   XXX.YYY.MED.VA.GOV           FLAGS:   Q
TRANSMISSION SCRIPT:  SCRIPT
TEXT:   X Q
```

Figure 9-1. Sample DOMAIN File (#4.2) entry

To perform this transfer (after it is properly set up), go into the UCI that has the messages that must be transmitted and invoke the XMNET-TRANSMISSION-MANAGEMENT menu. Choose the Send Messages to Another UCI option. You will then be prompted for the domain whose messages you wish to send. Be very careful, as there is no checking here and the messages in the queue are dumped and deleted from the queue before the receiver confirms recipients, etc. Answer with the name of the domain of the

## Sending and Receiving Messages Between UCIs

UCI you wish to send messages to. The messages will then be dumped into ^%ZISL and deleted from the queue.

When the transmission process is complete, go into the other UCI and invoke the Receive Messages from Other UCI option from the XMNET-TRANSMISSION-MANAGEMENT menu. The option selects the tasks in ^%ZISL for this UCI. When it finds a task destined for it, it will then read through and load what is in ^%ZISL (task number) into the Message file and deliver it to the correct recipients. Lastly, it will kill the ^%ZISL entry.

# 10. HELO Processing (XMROB)

## Overview

:  
Each successively smaller dot-piece of a network recipient's domain is first tested for an exact match in the DOMAIN file (#4.2). For example, ben.cs.umd.edu contains 4 dot-pieces. This prevents XXX.YYY.MIL from being matched by VA FileMan with MIL and MILWAUKEE (and possibly other MIL...s) and requiring users to choose from a list of matches. If no exact suffix match is found, the longest string of suffixes that is the prefix of at least one entry is accepted (with users choosing from a list of matches when appropriate). If no match can be made, the recipient is rejected and the "Domain not found." message is displayed.

To permit users to send responses (and new messages) to authors of received messages, if no suffix of a sending domain is defined, the rightmost (level 1) domain is added to the DOMAIN file (#4.2) (with routing through FORUM).

### Example:

If a transmission is received from AAA.JJJ.EEE, and neither JJJ.EEE nor EEE is in your DOMAIN file (#4.2), the following DOMAIN file (#4.2) entry will be automatically created for EEE.

```
NAME: EEE      FLAGS:      Q
RELAY DOMAIN: FORUM.VA.GOV
TRANSMISSION SCRIPT: AUTOTYPE:  OTHER
TEXT:
X      Q
```

**Figure 10-1. Sample DOMAIN file (#4.2 entry (with substituted data)**

A message is then sent to G.POSTMASTER of the receiving system to inform the system manager of the newly created domain. When such a message is received, the newly created DOMAIN file (#4.2) entry should be reviewed. Statistics will then be collected for it separately.

If you remove all synonyms to PSI.COM and make them domains, you will have much better statistics.



# Glossary

BANNER	A line of text with a user's name and domain, which is displayed to everyone who sends mail to the user.
BSCP	Block Mode Simple Communications Protocol, a procedure used for message transmission with error checking.
BULLETIN	A form letter often triggered by a VA FileMan field.
DEVICE	A terminal, printer, modem, or other type of hardware or equipment associated with a computer. A host file of an underlying operating system may be treated like a device in that it may be written to (e.g., for spooling).
DOMAIN	A site for sending and receiving mail.
INITIALIZATION	The process of setting variables in a program to their starting value.
INPUT TRANSFORM	An executable string of M code that is used to check the validity of input and converts it into an internal form for storage.
KEYWORD	A reference name that calls a help frame when entered at a message prompt.
LINE EDITOR	This is VA FileMan's special line-oriented text editor. This editor is used for the word-processing data type.
LOCAL	The system that a user is currently signed on to.
LOG IN/ON	The process of gaining access to a computer system.
LOG OUT/OFF	The process of exiting from a computer system.
MAIL BASKET	Mail baskets provide a way of saving messages in a sorted fashion similar to a filing system. Mail baskets are created at the "Message action" prompt by typing in "S" to save, then the name you wish to call the basket. If the basket already exists, the message will be sent to it. If the basket does not exist, you will be asked if you want it created. Placing a message in a mail basket other than the IN or Waste baskets protects the message from being automatically purged when the IN BASKET PURGE is run.
MESSAGE-ID	A message identifier that shows the time of transmission, the message number and the domain name of the message.
MULTIMEDIA MAIL	Multimedia Mail gives the capability of attaching Binary Large Objects (BLOBs) to electronic messages so that images, spreadsheets, graphs, and other operating system files that are not pure ASCII text, may be sent and received either locally or across the network.
PHYSICAL LINK DEVICE	Hardware used to establish outgoing communication.
"PLAYING A SCRIPT"	A method of opening a transmission link for a message, used to force message transmission of message and testing.

## Glossary

POLLER	An option that opens the transmission line to all domains with "P" in the Flags field.
POSTMASTER	The basket where message queues are stored. Also, the person who manages this basket for a particular site.
PROTOCOL	Code containing logic for opening and closing links, and for sending/receiving transmissions.
PURGE	A procedure used to delete messages or message pointers.
QUEUE	A list that stores messages destined for a given domain.
REMOTE	Any system that a user is not signed on to.
SCRIPT	A set of MailMan commands and transmission scripts to a remote domain in the DOMAIN file (#4.2).
SERVER	An automatic mail reader for internal messages.
SMTP	Simple Mail Transport Protocol. The primary transport protocol for MailMan.
SWP	Sliding Window Protocol.
TRANSMISSION SCRIPT	List of commands for directing a message stored in the TRANSMISSION SCRIPT field.
VALIDATION NUMBER	A security check number that must be in the domains of both the sending and receiving sites.
WRAP-AROUND MODE	Text that is fit into available column positions and automatically wraps to the next line, sometimes by splitting at word boundaries (spaces).



**REF:** For a comprehensive list of commonly used infrastructure- and security-related terms and definitions, please visit the ISS Glossary Web page at the following Web address:

<http://vaww.vista.med.va.gov/iss/glossary.asp>

For a comprehensive list of acronyms, please visit the ISS Acronyms Web site at the following Web address:

<http://vaww.vista.med.va.gov/iss/acronyms/index.asp>



# Appendix A—MailMan and TCP/IP

## Overview

This section describes how to set up a DEC VAX system to be able to use M programming language to communicate directly across a Transmission Control Protocol/Internet Protocol (TCP/IP) channel. When the steps in this process have been followed properly, MailMan Version 8.0 (the M communications software) will answer all requests for electronic mail service Simple Mail Transfer Protocol (SMTP) on the VAX's TCP/IP channel and will use this channel for sending mail to other TCP/IP SMTP hosts.

TCP/IP channels are very fast and require no application-level error checking. The error checking done at the physical layer of the protocol is Cyclical Redundancy Check with 16 bit result, also known as CRC-16, which is much better than that which we currently use at the application layer (usually Longitudinal Parity Checksum (LPC)). TCP/IP channels are also relatively immune to flow control problems. This method of message transmission and reception is therefore faster, freer from error, and more dependable. It is also a major protocol, namely the Internet protocol.

Sections following this Overview provide pertinent information, along with examples, that will allow system managers and programmers to effectively implement, support, and maintain MailMan's TCP/IP utilities and enhanced functionality.

## Using MailMan and TCP/IP

### Arriving

- A foreign TCP/IP SMTP mailer requests service on your TCP/IP channel.
- INET\_SERVERS, a job that runs under VMS intermediates this request that comes in on logical channel (socket) 25 and negotiates a channel (say, 1056) on which to actually do the work.
- When the actual channel on which the message transmission will take place has been decided, INET\_SERVERS will pass off the control of this channel (1056) to VistA MailMan via a VMS command procedure. INET\_SERVERS will then go back to monitoring socket 25 for the next service request.
- MailMan provides the SMTP receiver services to the remote (sending) e-mail system.
- When there is no mail left to send (or receive if the TURN command is executed) the logical channel is closed.
- System managers can input partial or full matches to network addresses into the NETWORK SENDERS REJECTED file (#4.501). When network mail receives a message from a sender that contains that string input, the sender will be rejected. This can be used to prevent large messages or large quantities of messages from coming to a system from a remote site.

## Sending

- MailMan will use the DEC supported open parameter to establish contact with a foreign CPU via a TCP/IP channel on socket 25 (where requests for SMTP service are made). This is done entirely in a transmission script. Though the OPEN appears to be opening a local device 25, it is really opening a connection to the remote host, while at the same time specifying SMTP service. The actual communications channel is negotiated. This means that multiple outgoing transmissions can be opened concurrently.
- After the channel has been successfully established, the transmission will continue using the SMTP protocol for data exchange.
- When there is no mail left to send (or receive if the TURN command is executed) the logical channel is closed.

## TCP/IP Channel

- This channel has as its physical layer (the lowest layer of any protocol) some synchronous means of transferring the data. This usually means X.25 or IEEE802.3 (Ethernet) is used.
- Error checking available through these protocols ensures that the stream of characters received is the same as sent (bus to bus). The most common method uses a checksum calculated via a method known as a CRC-16 (cyclical redundancy check with 16 bit result). This checksum is rated as very effective in detecting/eliminating transmission errors. The expected error rate is very low and is orders of magnitude better than the augmented LPC (longitudinal parity checksum) now used in VA MailMan.
- From the viewpoint of the M processes using TCP/IP channels, data is read and then parsed appropriately into segments. A special applications level protocol is necessary to do this.

```
INET_SERVERS => SERVERS.DAT => XMINET.EXE => XMINET.COM => MUMPS => MailMan
```

**Figure A-1. Receiver Flowchart**

The channel is passed to an M routine—XMRTCP.

The XMINET.INFO file is read by XMINET.EXE for user name and password. The password should not be known to anyone outside your site.

```
XMRTCP => ATCP X-REF OF DOMAIN FILE => XMCl (script interpreter) => XMS
```

**Figure A-2. Sender Flowchart (M Routines/Files)**

- Scripts are in the DOMAIN file (#4.2).
- XMS is the SMTP protocol driver and actually exchanges MPDUs (message protocol data units) with its counterpart on the other side of the connection.

## VMS Process Overview

These procedures will allow system managers to properly install, implement, and maintain MailMan with TCP/IP functionalities. This entire section discusses transmissions to other CPUs, creating VMS accounts, installing VMS files, and installing executable images.

## Setting up MailMan and TCP/IP

To set up MailMan with TCP/IP, do the following:

- Create a VMS account where the software is going to live. Make sure the owner is XMINET/PROT W:RE
- Install XMINET.COM, XMINET.EXE, XMINET.INFO and XMINET.C
- Create an EXECUTABLE image (invoked for the SMTP service)
- Edit TWG\$TCP:[NETDIST.ETC]SERVERS.DAT
- Install XMTCPQUE.COM in DUA0:[MUMPS]

The VMS portion of the set-up is exclusively for handling incoming mail. XMINET.EXE is invoked by INET\_SERVERS, the VMS process that handles all the connections from the network. XMINET.EXE sets up the correct environment so that a VMS M process can be invoked, and then it spawns off XMINET.COM. XMINET.COM then invokes a DSM environment with SOCKET^XMRTCP as the tied routine.

## M Process Overview

- Load XMRTCP & XMLTCP routines
- Set up entries in the following files:
  - COMMUNICATIONS PROTOCOL file (#3.4)
  - DEVICE file (#3.5)
  - TRANSMISSION SCRIPT file (#4.6)
  - TERMINAL TYPE file (#3.2)
- Make sure the site you want to connect to has MailMan or another SMTP mailer ready to receive your mail. Remember, you may only control to whom you send. You may advertise that you wish people to use this method to send you mail, but they are in control of their senders. If they have MailMan or another SMTP mailer ready to receive your mail, find out their IP address and place it in the MailMan Host field data (in the DOMAIN file (#4.2)) for that domain.
- Record the current DOMAIN file (#4.2) data for the domain you are about to change.

## Maintenance

- Above all, make sure the IP addresses you use are correct, check your queues, and test your connectivity regularly.
- The TCP Poller must be put into the DSM startup—TaskMan can not start it in most cases as TaskMan is not expected to be running (according to cookbook) on the node with the TCP/IP capability.
- A VMS process should try to restart the TCP Poller every 15 minutes or so. If it is already running, a duplicate name error will occur. This is normal. Better to have a duplicate name error (lets you know it is running) than not to have mail going out.
- If there is mail to go out and the Poller is running but the mail is not being sent out, FORCEXing the process for the TCP Poller and restarting it sometimes succeeds in making the system work again. It is not known why this works.

## Installation

- Stop the process INET\_SERVERS; this must be done from VMS.

```
$ sh sys
```

**Figure A-3. Stopping INET\_SERVERS**

Stop the process named "INET\_SERVERS".

- Restart the INET\_SERVERS process. In VMS this is done by typing in:

```
@twg$tcp:[netdist.misc]inetserv
```

**Figure A-4. Restarting INET\_SERVERS**

The following example illustrates the process described above.



**NOTE:** The domain used in the example that follows is FORUM. It can now receive TCP/IP mail at the IP address specified in the example given below.

Create a VMS Account:

- The account must have the SHARE privileges to use \_INETnn:
- The account must have the OPER privileges for debugging.
- The account must have a secure password that is recorded in XMINET.INFO

```

Username: XMINET                               Owner:  DHCP Mail->INET
Account:  NETWORK                             UIC:  [600,1153,50,153] ([XMINET])
CLI:     DCL                                  Tables: DCLTABLES
Default: VAL$:[XMINET]
LGICMD:  NL:
Flags:   DisCtly Restricted DisWelcome DisNewMail DisMail DisReport
Primary days:  Mon Tue Wed Thu Fri
Secondary days:  Sat Sun
Primary 000000000011111111112222 Secondary 000000000011111111112222
Day Hours 012345678901234567890123 Day Hours 012345678901234567890123
Network: ##### Full access #####           ##### Full access #####
Batch:   ----- No access -----           ----- No access -----
Local:   ----- No access -----           ----- No access -----
Dialup:  ----- No access -----           ----- No access -----
Remote:  ----- No access -----           ----- No access -----
Expiration: (none) Pwdminimum: 6 Login Fails: 0
Pwdlifetime: (none) Pwdchange: 13-MAR-1992 06:31
Last Login: (none) (interactive), 13-MAR-1992 10:05
(non-interactive)
Maxjobs: 0 Fillm: 150 Byt1m: 40960
Maxacctjobs: 0 Shrfillm: 0 Pbyt1m: 0
Maxdetach: 0 BI01m: 18 JTquota: 1024
Prclm: 2 DI01m: 18 WSdef: 900
Prio: 10 AST1m: 24 WSquo: 1200
Queprio: 0 TQElm: 10 WSextent: 1500
CPU: (none) Enqlm: 100 Pgf1quo: 10240
Authorized Privileges:
TMPMBX OPER NETMBX SHARE
Default Privileges:
TMPMBX OPER NETMBX SHARE
Identifier Value Attributes
DSM$MANAGER_DSMMGR %X80010004

```

Figure A-5. Sample VMS account



**CAUTION:** Be sure to allow **ONLY** Network access!

Install VMS Files:

- Put XMINET.COM in the above account's directory.
- Put XMINET.EXE in twg\$tcp:[netdist.serv].
- Put XMINET.INFO in twg\$tcp:[netdist.serv].

## Create and Install Executable Image

This executable image has been compiled and linked from the C program file XMINET.C shown below. The executable image has been put into XMINET.EXE.



**CAUTION: XMINET.EXE is installed in twg\$tcp:[netdist.serv]. and reads XMINET.INFO—this is for secure passwords.**

```

/* Create network jobs through loginout - many thanks to the Wollongong.
**      Note: the use of CREPRC is documented by DEC as 'reserved for
**      their use' so we use it with caution.
xxx@kansas-city.va.gov
**
** This account should have SHARE priv by default so it can access the
** _INETnn: device at the same time as the creating process.
**
*/

#include <stdio.h>
#include <descrip.h>
#include <ssdef.h>
#include <prcdef.h>
#include <dvidf.h>

/*
* The user information for a network job is passed in the 'output'
* parameter to $CREPRC. The format of this buffer is as follows:
* A two byte flag word followed by ASCII strings containing the
* username, password, and account information.
*
* Maximum buffer sizes for the various pieces in this block are as
* follows:
*/

#define AMAX 20                /*maximum account*/
#define BMAX 80                /*scratch buffer max*/
#define PMAX 31                /*maximum password*/
#define UMAX 12                /*maximum username*/
#define MAXBUF (2 + 1 + AMAX + 1 + UMAX + 1 + PMAX)

#define NO_ERROR 0
#define ERROR 1

/* function prototypes */

long int refs(char *device);
int readuname(char *username, char *password);

main ( int argc, char **argv )
{
    char user_info[MAXBUF], *cp, sysnet[PMAX], username[UMAX];
    char password[PMAX];
    int sts,i,rcount;
    int unamelen, passlen, accountlen;
    int stsflg = PRC$M_DETACH | PRC$M_NETWRK;

    $DESCRIPTOR(input_d, "XMINET.COM"); /*input must be a .COM file */
    $DESCRIPTOR(image_d, "SYS$SYSTEM:LOGINOUT.EXE");
    $DESCRIPTOR(output_d, user_info);
    $DESCRIPTOR(sysnet_d, sysnet);

    /* read login info from password file */
    if (readuname(username,password)) exit(ERROR);

    output_d.dsc$w_length = MAXBUF;
    unamelen = strlen(username);
    passlen = strlen(password);

```

```

cp = user_info;
*cp++ = '\0';           /*zero flag word tells loginout */
*cp++ = '\0';           /*not to try proxy validation */

/*Username in ASCII format*/
*cp++ = (char)unamelen;
for (i =0; i < unamelen; *cp++=username[i++]);

/*Password in ASCII format*/
*cp++ = (char)passlen;
for (i =0; i < passlen; *cp++=password[i++]);

/*Account information in ASCII format
*      -NOT USED HERE-      */
accountlen = 0;
*cp++ = accountlen;

output_d.dsc$w_length =
(2+1+unamelen+1+passlen+1+accountlen);

fgetname(stdout,sysnet);      /* pass the INETnnn device name */
sysnet_d.dsc$w_length=strlen(sysnet);

rcount=refs(sysnet);          /*current ref count before qio*/

sts = sys$creprc(0,           /* PID           */
&image_d,                    /* Image to Run */
&input_d,                     /* Input         */
&output_d,                    /* Output        */
&sysnet_d,                    /* Error         */
0,                             /* Priv's        */
0,                             /* Quota         */
0,                             /* Process Name  */
0,                             /* Priority      */
0,                             /* UIC           */
0,                             /* Term MBX     */
0,                             /* Net/Detach   */
stsflg);

/* Probably should check a mailbox for status of created process
*/

    if (!(sts & 1)) lib$stop(sts);

/* Check the refcount on the mailbox device, 60 seconds is all we are allowing
* for the new process to get created and run the .com procedure.  If the
* refcount changes then the detached process has accessed it and we can just
* hang around until the new process is done with it.  When we exit this one
* the Master server will gracefully close the sockets and delete the INET
* device.
*/

    for (i=0; i<60; i++) {           /* look for new process ref */
        if (rcount != (refs(sysnet))) break;
        else sleep(1);
    }

    if ( i == 59 ) exit(ERROR);      /* process never got created */
                                    /* so we just exit          */

    for (;;) {                       /* Look for new process to exit - refcount drops */
        if (rcount == (refs(sysnet))) break;
        else sleep(5);
    }

    exit(ERROR);                     /* exit - signals master */
                                    /* server to close up shop */
}

long int refs(char *device)          /*returns reference count for a device*/
{
    long int refcount,status;

struct      /* item list for one item code */
{

```

```

    short buffer_len; /* length of buffer that getdvi writes into */
    short item_code; /* symbolic code of information requested */
    long buf_addr; /* address of buffer that getdvi writes into */
    long ret_len_addr; /* number of bytes getdvi writes into buffer */
    long terminator; /* longword to terminate item list
*/
    } itmlst;
struct /* io status block */
{
    long cond_val; /* condition value returned */
    long not_used; /* from the getdvi service */
    } iosb;

/* build descriptor for the device name parameter */
$DESCRIPTOR (dev_name_desc,device);
dev_name_desc.dsc$w_length=strlen(device);

/* set up item list */
itmlst.buffer_len = 4; /* length of longword (refcount) */
itmlst.item_code = DVI$_REFCNT; /* reference count */

itmlst.buf_addr = &refcount; /* address of buffer to put result */
itmlst.ret_len_addr = 0; /* not used */
itmlst.terminator = 0; /* must have longword of 0
to terminate item list
*/

/* call system service */
status = sys$getdviw(0,0,&dev_name_desc,&itmlst,&iosb,0,0,0);

/* check return values */

if (! (status & 1)) lib$stop (status);
if (! (iosb.cond_val)) lib$stop (status);

return(refcount);
}

/* Requires a password file 'twg$tcp:[netdist.serv]xminet.info'
** and the format must be any number of comment lines beginning with a #,
** followed by a line which contains username^password, an ^ delimited string.
*/

int readuname(char *username, char *password)
{
    FILE *pwdfile;
    char buf[81];
    int i,j;

/* read from the password file */
if ((pwdfile = fopen("twg$tcp:[netdist.serv]xminet.info", "r")) == NULL) {
    fprintf(stderr,"Could not open password file 'xminet.info'.\n");
    return (ERROR);
}

while (fgets(buf, 80, pwdfile) != NULL) {
    if (strncmp(buf,"#",1)) {
        for (i=0; ((buf[i] != '^') && (i < 80 )); i++) {
            username[i]=buf[i];
        }
        username[i++]='\0';

        for (j=0; (j < (80-strlen(username)) && (buf[i]>32)); j++) {
            password[j]=buf[i++];
        }
        password[i]='\0';

        fclose(pwdfile);
        return (NO_ERROR);
    }
}
fclose(pwdfile);
return (ERROR); /* if we get here, there was nothing to read */

```



```
}

```

Figure A-6. Sample executable image

## XMINET.COM

XMINET.EXE invokes the following com file that is installed in the directory VA1\$:[XMXMINET].



**NOTE:** To troubleshoot, look at inetmm.log and inetmm.debug. The file (inetmm.debug) is created if debug is set in inetmm.com.

```

$!XMINET.COM - Called from the $CREPRC system service in inetmm.c
$! as the input parameter. Parse the logical SYS$NET which contains
$! the INETnn mailbox device to read/write through. This value gets
$! passed to SYS$NET through the error parameter of the $CREPRC.
$!-----
$!
$! def sys$output inetmm.debug      !This is the debug log
$! debug=1                        !Set to 1 to turn on debug mode
$! set noon                        !Don't stop
$! set process/prio=1              !Keep things in check
$! set proc/priv=(share)           !Required to use the MBX device
$! if debug
$! then
$! show logical/all
$! reply/enable/temp
$! reply/user=UUUUUUUU "From the XMINET server -->IN."
$! else
$! set nover                       !Turn off verify
$! endif
$! z=f$trnlnm("sys$net")           !This is our MBX device
$! x=f$extract(0,f$locate(":",z)+1,z)
$! write sys$output x              !This can be viewed in the log file
$! set nover                       !Don't pass this stuff to the output device
$! define/process sys$output 'x'
$! define/process sys$input 'x'
$! define/process sys$command 'x'
$!-----
$! **Be sure this command line is correct for your system
$! **and if access control is enabled that this account has
$! **access to this uci,vol & routine.
$! dsm/envirn=UUUUUUUU/uci=XXX/vol=###/data="'x'" SOC25^XMRTCP
$!-----
$! if debug then -      reply/user=UUUUUUUU "From the XMINET server -->OUT."
$! logout/brief.

```

Figure A-7. Sample XMINET.COM file

The two occurrences of UUUUUUUU should be changed to the username for the M configuration this software will work with. On the line where DSM is invoked at SOC25^XMRTCP, the UCI (XXX) and vol (###) should be edited as well.



**CAUTION:** XMINET.INFO holds your password information for XMINET.EXE to read.

Assign a password to your XMINET account. Make sure that the password that is recorded in this file is correct. It will be read by XMINET.EXE. This way, people that know how this scenario works may not log into this account to do anything other than mail.

In the example below, change INETMM^INETMM to CAPTAIN^MIDNIGHT if your username was changed to CAPTAIN and your password was changed to MIDNIGHT for the VMS account.

```
#xmdhcp.info [really in xminet.info] - this file must contain a
non-commented ([non-]#) line with the
# USERNAME^PASSWORD corresponding to the VMS account for use with
TCP/IP
# dhcp mailman:
#
INETMM^INETMM
#
#
```

**Figure A-8. XMINET.INFO routine**

### Edit TWG\$TCP:[NETDIST.ETC]SERVERS.DAT

First, do the following:

```
copy twg$tcp:[netdist.etc]servers.dat servers_wolly.dat
```

The command above will preserve the original Servers.Dat file as servers\_wolly.dat, so that you may see or return to original values if it becomes necessary. Renaming servers\_wolly.dat will put the TCP/IP INET\_SERVERS routine back to using the original Wollongong supplied servers (including the SMTP server).

Edit TWG\$TCP:[NETDIS.ETC]SERVERS.DAT so that it will invoke XMINET.EXE that will invoke MailMan whenever mail is received.

The Servers.Dat file includes entries for all the services supported by Wollongong. Look for the one titled #SMTP like the one below (see [Figure A-9](#)). Each entry has a series of fields. The second field in the entry is the executable image that will be invoked when SMTP service is requested. It will say "TWG\$TCP:[NETDIST.SERV]SMTP.EXE" if it is the one released by Wollongong originally. Change it to TWG\$TCP:[NETDIST.SERV]XMINET.EXE.

The following is an example of what the file should look like with your change:

```
#SMTP
service-name      Smtp
program           TWG$TCP:[NETDIST.SERV]XMINET.EXE
socket-type       SOCK_STREAM
socket-options    SO_ACCEPTCON | SO_KEEPAIVE
socket-address    AF_INET , 25
working-set      1024
maxservers        4
priority          4
INIT              TCP_Init
LISTEN            TCP_Listen
CONNECTED         TCP_Connected
SERVICE          Run_Program
```

**Figure A-9. Sample TWG\$TCP:[NETDIST.ETC]SERVERS.DAT file**

## M/VMS Software

The XMRTCP routine *must always be running* on your system on the node that the TCP/IP channel (X.25 board) is physically on.

This M routine usually cannot be scheduled through TaskMan. This is because TaskMan does not usually run on this particular node. It must be scheduled to run from VMS. XMTCPQUE.COM must be invoked to start it running. It should be scheduled to run from VMS every hour or so. If there is already one running, the new one will quit.

```

$! XMTCPQUE.COM -
$! FOR FORMII 6/24/92 TO START UP POLLER FOR DELIVERY OF TCP/IP SMTP
$! mail - all domains with TCPCHAN FLAG set (usually stored in DUA0:[MUMPS]
$ set noon
$ dsm/env=eeeeee/uci=uuu/vol=vvv START^XMRTCP
$ exit

```

**Figure A-10. Scheduling the XMRTCP to run by invoking XMTCPQUE.COM**

## M Software

### Routines in MailMan V. 8.0

#### XMRTCP

The XMRTCP routine is run in the background to start transmissions to domains that have their TCP/IP Transmission Flag set. It is also invoked at another tag to initialize the receiver that is invoked by INET\_servers.

#### XMLTCP

The XMLTCP routine is the guts of the send/receive protocol and reads the TCP/IP channel. It is invoked in XMR\* and XMS\* when the communications protocol is TCP/IP-MAILMAN (in the COMMUNICATIONS PROTOCOL file [#3.4]).

## Other Mandatory Records

```

NAME: TCP/IP-MAILMAN
SEND: D SEND^XMLTCP ; I XM["D" S XMTRAN="S: "_$_E(XMSG,1,242) D TRAN^XMC1 RECEIVE:
D REC^XMLTCP ; I XM["D" S XMTRAN="R: "_$_E(XMRG,1,242) D TRAN^XMC1 OPEN:
S (X,XMLINE,XMLCHAR,XMLER,XMLST)=0,XMLTCP=" ",XMNO220=" ",XMSTIME=
180 X ^%ZOSF("RM")
CLOSE: K XMLTCP L -^XMBX("TCPCHAN",XMHOST) Q
DESCRIPTION: Protocol for MailMan to use across a TCP/IP channel.

```

**Figure A-11. Sample COMMUNICATIONS PROTOCOL file (#3.4) entry**

```

NAME: NULL DEVICE                                $I: _NLA0:
VOLUME SET(CPU): GGL                             SIGN-ON/SYSTEM DEVICE: NO
LOCATION OF TERMINAL: NULL                        MARGIN WIDTH: 132
FORM FEED: #                                       PAGE LENGTH: 64
BACK SPACE: $C(8)                                  SUBTYPE: P-DEC
TYPE: TERMINAL
    
```

Figure A-12. Sample DEVICE file (#3.5) entry

```

NAME: TCPCHAN-SOCKET25
TEXT:
X S XMRPORT=25,X="ERRSCRIPT^XMRTCP",@^%ZOSF("TRAP")
X L +^XMBX("TCPCHAN",XMINST):3 E S ER=1,XMER="CHANNEL IN USE"
X S XMSIO="XMRPORT:(TCPCHAN,ADDRESS=XMHOST):30"
X O @XMSIO E S ER=1 L -^XMBX("TCPCHAN",XMHOST)
X U XMRPORT:NOECHO
X S XMHANG="C XMRPORT S IO=""",IOST(0)=0",IO=XMRPORT
L:180 220 MAIL
X L -^XMBX("TCPCHAN",XMHOST) K XMSIO
X S IO=IO(0) X ^%ZOSF("EON")
    
```

Figure A-13. Sample TRANSMISSION SCRIPT file (#4.6) entry

```

NAME: FORUM.VA.GOV                                FLAGS: T
MAILMAN HOST: 1.0.0.1                            DISABLE TURN COMMAND: NO
PHYSICAL LINK DEVICE: NULL DEVICE
TRANSMISSION SCRIPT: SCRIPT TEXT:
O H=FORUM.VA.GOV,P=TCP/IP-MAILMAN
C TCPCHAN-SOCKET25
TCP/IP POLL FLAG: POLL
    
```

Figure A-14. Sample DOMAIN file (#4.2) entry

```

NAME: P-DEC                                       SELECTABLE AT SIGN-ON: YES
RIGHT MARGIN: 132                                FORM FEED: #
PAGE LENGTH: 64                                  BACK SPACE: $C(8)
OPEN EXECUTE: W *27,"[w"                        10 PITCH: $C(27)"[w"
12 PITCH: $C(27)"[2w"                            RESET: $C(27)"[w"
DESCRIPTION: general DEC printer
    
```

Figure A-15. Sample TERMINAL TYPE file (#3.2)





# Appendix B—TCP/IP Poller

## Overview

In order to send out mail, MailMan usually sets up a task and schedules it to run on the node of the cluster that has the device associated with the job. This will work if TaskMan is running on the node that has TCP/IP capability. Simply set up a device with `_NLA0:` (the null device) as the `$I` and enter the correct data into the fields for Volume and UCI. Then put this device into the Physical Link field of the domain you are sending to TCP/IP mail. It will now work just like all the other "normal" domains. Make sure the other fields in the DOMAIN file (#4.2) are correct: `FLAGS`, `TRANSMISSION SCRIPT`.

However, in the usual VistA VAX configuration, the node of the cluster that has TCP/IP capability does not run TaskMan. (This node is simply not used for background processing.)

If you do not have TaskMan running on the cluster node that has TCP/IP capability you must do the following:

- Set up a VMS com file that invokes `POLL^XMRTCP`.
- Schedule the above VMS com file to run regularly (every nn minutes).
- The job above will produce an error each time it runs if the poller is already running—this is normal.
- Flag each domain that you send mail to using the `TCP/IP-SOCKET25` protocol by entering `POLL (1)` into the `TCP/IP POLL FLAG` field.

`POLL^XMRTCP` runs every 60 seconds and tests the queues in the DOMAIN file (#4.2) that have the `TCP/IP POLL FLAG` set. It is single-threaded. It will only send one queue at a time. Then it will look for another queue that has messages to send and make the proper connection to send that queue. Mail should go out very quickly. The protocol has been clocked at line speeds of 2000 characters/second on a 19.6 line and 40,000 characters/second on a Local Area Network (LAN via Ethernet).

If the queues seem to back up you can take the following actions that will usually work.

- Call the site you are not able to open a connection to. It is possible that a connection was dropped to this site and still is running there. They will have to force exit the process. It will be called `INETnnnn`, where `nnnn` represents the socket number that the exchange was renegotiated to run on. One error code you will get when you try to open a connection under these circumstances is 61.
- Use the DSM utility `^%SY`. If a job named `MM-TCP-Poller` is not running, start it. (The poller explicitly set its name to `MM-TCP-Poller`.)
- If the poller is running, monitor it to see if it seems to be doing anything. If it is not effectively sending out any mail, `FORCEX` the process and restart it.

How do you tell if the poller is *effectively* running? Look at the job using the `%JOB` utility supplied by DSM. If the job is not in a `HANG` state, list the variables, wait 60 seconds, and check the queue again (tag `QQ`). If the counts for `Cmnds`, `Refs` & `DIO` do not change for a long time (90 seconds or so), it is not doing anything.

- If the above didn't work, look in the error trap or call your local ISC.

## Setting up More Domains to Use the TCP Channels

The steps to setting up more domains to use TCP channels are much simpler than the first. You need to do the following:

- Make sure the site you want to connect to has MailMan or another SMTP mailer ready to receive your mail. Remember, you may only control who you send to this way. You may advertise that you wish people to use this method to send you mail, but they are in control of their senders. If they have MailMan or another SMTP mailer ready to receive your mail, find out their IP address. It will become the MailMan Host field data in the DOMAIN file (#4.2) for that domain.
- Record the current DOMAIN file (#4.2) data for the domain you are about to change so it looks like the one below:

```

NAME: FORUM.VA.GOV                FLAGS: T
MAILMAN HOST: 1.0.0.1             DISABLE TURN COMMAND: NO
TCP/IP POLL FLAG: POLL           PHYSICAL LINK DEVICE: NULL DEVICE
TRANSMISSION SCRIPT: SCRIPT
TEXT:
O H=FORUM.VA.GOV,P=TCP/IP-MAILMAN
C TCPCHAN-SOCKET25

```

Figure B-1. Sample DOMAIN file (#4.2)



### CAUTION:

- Make sure that the **FLAGS** field does not have an **S** in it unless you have TaskMan running on the same machine as your TCP/IP is located. If there is an **S** in the **FLAGS** field, a task is created to deliver the mail in the queue and it will fail with many errors because the TCP/IP device is not available.
- The **TCPCHAN FLAG** field must be set to **TCPCHAN POLLING REQUIRED**. This is because a job runs in the background (XMRTCP must be started manually on the node where the TCP/IP channel resides) that decides which domains with messages queued to be sent, should be sent via the TCP/IP channel. It will ignore all domains that do not have this flag set and the messages will stay in the queue indefinitely without being sent.



# Index

## A

^XMRTCP Routine, 7-2

## 1

1SCP Protocol, 2-5  
1SCP Protocols, 5-2

## 3

3BSCP, 2-4, 2-5, 2-6, 5-1  
3BSCP Protocol, 2-4, 2-5, 5-2

## A

ACK, 5-1, 5-2  
Acronyms (ISS)  
    Home Page Web Address, Glossary, 2  
Actively Transmitting/Receiving Queues Report Option, 1-7  
Adobe Acrobat Quick Guide Web Address, xvi  
Adobe Home Page Web Address, xvi  
Allow Site to Reject Network Mail from Particular Senders, 1-3  
Answering a Message Internet Style, 1-4  
Appendix A—MailMan and TCP/IP  
    A, 1  
    Overview  
        A, 1  
Appendix B—TCP/IP Poller  
    B, 1  
    Overview  
        B, 1  
Application Layer, 3-1  
    Overview, 3-1  
Arriving Basket, 4-4  
Assumptions About the Reader, xv  
Asynchronous Communications, 2-2  
Avoiding Infinite Transmission Loops, 6-1

## B

Binary Large OBjects, 1-1, 2-7  
BLOB, 1-2  
BLOBs, 2-7  
Block Mode, 2-4, 5-2  
Block Mode (3BSCP) Protocols, 5-2  
broadcast messages, 4-4

## C

CALL Command, 3-11, 3-14

Callout Boxes, xiv  
Case Sensitive Message Fields from Network Receptions, 1-3  
Changes to VistA Domains, 6-1  
Channels  
    DECNET, 8-1  
Christen a domain Option, 1-6  
Christen a Domain Option, 1-5  
CLOSE EXECUTE Field, 2-3  
Commands  
    CALL, 3-11, 3-14  
    DIAL, 3-11, 3-15  
    E, 1-2, 3-7  
    ERROR, 3-11  
    FLUSH, 3-11, 3-14  
    LOOK, 3-11, 3-16, 3-17  
        Original, 3-12  
        Updated, 3-12  
    LOOK 220, 3-16  
    MAIL, 3-11, 3-16  
    Network Script, 3-10  
    OPEN, 3-11, 3-16  
    Script, 3-10  
    SEND, 3-11  
    WAIT, 3-11, 3-14  
    XECUTE, 3-11, 3-16  
Communication Protocols, 5-2  
COMMUNICATIONS PROTOCOL File (#3.4), 2-1, 2-4, 3-11, 8-1, 3  
Compare Domains in System Against Released List Option, 1-6  
Components  
    Network MailMan, 2-1  
Create a LOGIN.COM File on Your Local Machine, 8-2  
Create an Account on the Remote Host, 8-4  
Create and Install Executable Image  
    A, 6  
Create the LOGIN.COM File, 8-5

## D

Data Dictionary  
    Data Dictionary Utilities Menu, xv  
    Listings, xv  
DCL Context, 7-2  
DECNET, 8-1, 8-2, 8-5  
    Channel, 8-1  
    Node, 8-1  
    Task, 8-1  
Define the Remote DECNET Node on the Remote Machine, 8-6  
Define the Remote DECNET Node on Your Machine, 8-3  
DEVICE File (#3.5), 2-1, 2-2, 2-3, 5-4, 8-2, 3  
Devices, 2-2, 4-3  
DIAL command, 3-15  
DIAL Command, 3-11, 3-15

## Index

Directory of Remote Recipients (MailLink) Expanded, 1-4  
Display message queues Option, 1-7  
Documentation  
    Revisions, iii  
    Symbols, xiii  
DOMAIN File (#4.2, 2-1)  
DOMAIN File (#4.2), 1-2, 1-3, 1-5, 1-6, 2-1, 3-6, 4-1, 4-2,  
    6-1, 6-2, 6-3, 7-1, 8-1, 8-2, 9-1, 10-1, 2, 3, 1, 2  
    Glossary, 2  
domain name, 3-1  
Domain Name, 4-2  
    Management, 4-1  
    Resolution, 4-2  
Domain Name Server, 2-1  
Domain names, 4-1  
Domain Names, 1-3, 4-1, 6-1, 9-1  
Domains, 1-5, 2-1, 6-1

## E

E Command, 1-2, 3-7  
Edit a script Option, 1-9  
Enter a Transmission Script in the Domain File, 8-2  
Error Checking  
    Communications Protocols, 2-4  
ERROR Command, 3-11, 3-16  
error message, 1-2  
Ethernet, 1-2, 7-1, 2, 1  
EVS Anonymous Directories, xvii  
Executable Image, 6

## F

Faster NETMAIL Transmission Speeds using TCP/IP  
    Channels, 1-3  
Features  
    Network MailMan, 1-2  
Fields  
    CLOSE EXECUTE, 2-3  
    OPEN TIME-OUT, 2-3  
    TRANSMISSION SCRIPT, 6-1  
    VALIDATION NUMBER, 4-2  
    VOLUME SET(CPU) Field, 4-3  
Figures and Tables, ix  
File Transmission Protocol, 1-5  
Files  
    COMMUNICATIONS PROTOCOL (#3.4), 2-1, 8-1, 3  
    COMMUNICATIONS PROTOCOL(#3.4), 2-4, 3-11  
    DEVICE (#3.5), 2-1, 2-2, 2-3, 5-4, 8-2, 3  
    DOMAIN (#4.2, 2-1)  
    DOMAIN (#4.2), 1-2, 1-3, 1-5, 1-6, 2-1, 3-6, 4-1, 4-2,  
        6-1, 6-2, 6-3, 7-1, 8-1, 8-2, 9-1, 10-1, 2, 3, 1, 2  
        Glossary, 2  
    DOMAIN [#4.2], 3  
    INTER-UCI TRANSFER (#4.281, 2-1)  
    LOGIN.COM, 8-1, 8-2  
    MAILBOX (#3.7), 2-1  
    MAILMAN SITE PARAMETERS (#4.3), 1-4, 2-1, 3-7,  
        4-4  
    MAILMAN TIME ZONE (#4.4), 2-1

MESSAGE DELIVERY STATS (#4.2998, 1-2, 2-1)  
MESSAGE STATISTICS (#4.2999), 2-1  
NETWORK SENDERS REJECTED (#4.501), 2-1, 3-2,  
    1  
NEW PERSON (#200), 1-3  
REMOTE USER DIRECTORY (#, 2-1)  
REMOTE USER DIRECTORY (#4.2997), 1-4  
TERMINAL TYPE (#3.2, 2-1, 2-2, 2-3, 5-4, 3)  
TRANSMISSION SCRIPT (#4.6, 2-1, 3-14, 3)  
XMINET.INFO  
    A, 2  
FLUSH Command, 3-11, 3-14  
FORUM, iv, 3-5, 3-6, 4-1, 4-2, 6-2, 7-1, 10-1, 5  
Forwarding Addresses Automatically Verified, 1-3  
FTP, xvii, 1-2, 1-5, 2-7, 7-1, 7-2

## G

Glossary, 1  
    ISS Home Page Web Address, Glossary, 2

## H

HELO Processing (XMROB), 10-1  
    Overview, 10-1  
Help  
    At Prompts, xv  
    Online, xv  
    Question Marks, xv  
Historical Queue Data/Stats Report Option, 1-7  
Home Pages  
    Adobe Acrobat Quick Guide Web Address, xvi  
    Adobe Web Address, xvi  
    HSD&D Home Page Web Address, xvi  
    ISS  
        Acronyms Home Page Web Address, Glossary, 2  
        Glossary Home Page Web Address, Glossary, 2  
    MailMan Home Page Web Address, xvi  
    VistA Documentation Library (VDL) Home Page Web  
        Address, xvi  
How to  
    Obtain  
        Technical Information Online, xiv  
    Route Internet Traffic through FORUM, 7-1  
    Use a DECNET Channel, 8-1  
    Use this Manual, xiii  
How To  
    Use this Manual, xiii  
HSD&D  
    Home Page Web Address, xvi  
Hunt Group, 4-3

## I

IDCU, 3-7, 3-8, 7-2  
images, 1-2  
Instructions  
    Completing a DECNET Channel, 8-1  
Internet, 7-1  
INTER-UCI TRANSFER File (#4.281, 2-1)

- Introduction
  - MailMan Network Reference Manual, 1-1
- ISS
  - Acronyms
    - Home Page Web Address, Glossary, 2
  - Glossary
    - Home Page Web Address, Glossary, 2
- Issue Appropriate Commands, 8-5
  
- K**
- kill transmission, 4-4
  
- L**
- LAN, 1-2, 1
- List a transcript Option, 1-7
- List File Attributes Option, xv
- Local Area Network, 1-2, 1
- LOGIN.COM File, 8-1, 8-2
- Longitudinal Parity Checksum, 2-5, 5-2, 1
- LOOK 220 Command, 3-16
- LOOK Command, 3-11, 3-16, 3-17
  - Original, 3-12
  - Updated, 3-12
- LPC, 1-3, 2-5, 5-1, 5-2, 1, 2
  
- M**
- MAIL Command, 3-11, 3-16
- Mail Name, 1-3
- MAILBOX File (#3.7), 2-1
- MailLink, 1-4
- MailMan
  - Menu Structure, 1-6
- MailMan Home Page Web Address, xvi
- MAILMAN SITE PARAMETERS File (#4.3), 1-4, 2-1, 3-7, 4-4
- MAILMAN TIME ZONE File (#4.4), 2-1
- Management Overview, 2-1
- Menus
  - Data Dictionary Utilities, xv
  - MailMan Menu Structure, 1-6
  - Network Management, 1-4, 1-6
  - Queue Management, 1-5, 1-6
  - Transmission Management, 9-1
  - Transmission Management menu, 1-8
  - Transmission Management Menu, 1-5
  - XMNET, 1-4, 1-6
    - Functional Description, 1-4
  - XMNETE-TRANSMISSION-MANAGEMENT, 1-8
  - XMNET-QUEUE-MANAGEMENT, 1-5, 1-6
  - XMNET-TRANSMISSION-MANAGEMENT, 1-5, 9-1
- Message Delivery Statistics, 1-2
- MESSAGE DELIVERY STATS File (#4.2998), 1-2, 2-1
- Message Protocol Data Units, 2-6
- MESSAGE STATISTICS File (#4.2999), 2-1
- Mini Out Device, 2-3
- Modems, 2-2, 3-15, 5-4, 1
  - Support, 5-4
- MPDUs, 2-6, 2
- Multimedia Mail, 1-1, 1-2
  
- N**
- NAK, 5-1, 5-2
- Name Server, 6-1
- NETMAIL, 1-2, 1-3
- Network Architecture, 2-1
  - Overview, 2-1
- Network Delivery Script Error (E Command), 1-2
- Network File System, 1-2
- Network Help Option, 1-6
- Network Layer, 4-1
  - Overview, 4-1
- Network Mail
  - TCP/IP, 7-1
    - Overview, 7-1
  - Troubleshooting, 6-1
    - Overview, 6-1
- Network Mail Across a DECNET Channel, 8-1
  - Overview, 8-1
- Network MailMan, 2-1
- Network MailMan Features, 1-2
- Network Management Menu, 1-4, 1-6
- Network Script Commands, 3-10
- Network Scripts, 3-7
- Network Senders Rejected, 3-2
- NETWORK SENDERS REJECTED File (#4.501), 2-1, 3-2, 1
- NEW PERSON File (#200), 1-3
- NFS, 1-2
- Nodes
  - DECNET, 8-1
- Notes on the Mini Out Device, 2-3
  
- O**
- Online
  - Documentation, xv
  - Technical Information, How to Obtain, xiv
- OPEN Command, 3-11, 3-16
- OPEN TIME-OUT Field, 2-3
- Options
  - Actively Transmitting/Receiving Queues Report, 1-7
  - Christen a domain, 1-6
  - Christen a Domain, 1-5
  - Compare Domains in System Against Released List, 1-6
  - Data Dictionary Utilities, xv
  - Display message queues, 1-7
  - Edit a script, 1-9
  - Historical Queue Data/Stats Report, 1-7
  - List a transcript, 1-7
  - List File Attributes, xv
  - MailMan Menu Structure, 1-6
  - Network Help, 1-6
  - Network Management, 1-4, 1-6
  - Play a script, 1-9
  - Queue Management, 1-5, 1-6
  - Queues with Messages to Transmit Report, 1-7, 3-5

## Index

- Receive Messages from Another UCI, 1-9
- Send Messages to Another UCI, 1-9
- Sequential Media Message Reception, 1-9
- Sequential Media Queue Transmission, 1-10
- Show a queue, 1-8
- Site Parameters, 1-5, 1-8
- Subroutine editor, 1-10
- Transmission Management, 9-1
- Transmission Management menu, 1-8
- Transmission Management Menu, 1-5
- Transmit a Single Queue, 1-8
- Transmit All Queues, 1-8
- Validation Number Edit, 1-10
- XMCHRIS, 1-5, 1-6
- XMEDIT-DOMAIN-VALIDATION#, 1-10
- XMHIST, 1-7
- XMLIST, 1-7
- XMNET, 1-4, 1-6
  - Functional Description, 1-4
- XMNETE-TRANSMISSION-MANAGEMENT, 1-8
- XMNETHELP, 1-6
- XMNET-QUEUE-MANAGEMENT, 1-5, 1-6
- XMNET-TRANSMISSION-MANAGEMENT, 1-5, 9-1
- XMQACTIVE, 1-7
- XMQDISP, 1-7
- XMQDOMAINS, 1-6
- XMQSHOW, 1-8
- XMQUEUED, 1-7
- XMR-SEQ-RECEIVE, 1-9
- XMR-UCI-RCV, 1-9
- XMR-UCI-SEND, 1-9
- XMSCRIPTEDIT, 1-9
- XMSCRIPTPLAY, 1-9
- XMSITE, 1-5, 1-8
- XMS-SEQ-TRASMIT, 1-10
- XMSTARTQUE, 1-8
- XMSTARTQUE-ALL, 1-8
- XMSUBEDIT, 1-10
- Orientation, xiii
- Other Mandatory Records
  - A, 12
- Overview
  - Appendix A—MailMan and TCP/IP
    - A, 1
  - Appendix B—TCP/IP Poller
    - B, 1
  - Application Layer, 3-1
  - HELO Processing (XMROB), 10-1
  - Network Architecture, 2-1
  - Network Layer, 4-1
  - Network Mail
    - TCP/IP, 7-1
    - Troubleshooting, 6-1
  - Network Mail Across a DECNET Channel, 8-1
  - Sending and Receiving Messages Between UCIs, 9-1
  - TCP/IP, 7-1
  - Transport Layer, 5-1
  - Troubleshooting, 6-1
- P**
  - Patches
    - Revisions, iv
  - Play a script Option, 1-9
  - Playing a Script, 3-7, 4-3
  - Poller
    - TCP/IP, 7-2
  - Postmaster, 1-7, 1-8, 3-6, 4-1, 4-3, 4-4
  - POSTMASTER, 10-1
  - Process
    - Transmission, 4-2
  - Progress Report Logic, 4-4
  - protocols, 2-4
  - Protocols
    - 1SCP, 2-5, 5-2
    - 3BSCP, 2-4, 2-5, 5-2
    - Block Mode (3BSCP), 5-2
    - Communication, 5-2
    - SCP, 2-4, 5-2
    - SWP, 2-4, 2-5, 5-2
    - Test, 2-6
- Q**
  - Query Display After Message Has Been Sent, 3-9
  - Question Mark Help, xv
  - queue, 4-4
  - Queue Management Menu, 1-5, 1-6
  - Queues, 3-5, 4-3
  - Queues with Messages to Transmit Report Option, 1-7, 3-5
- R**
  - Reader, Assumptions About the, xv
  - Receive Messages from Another UCI Option, 1-9
  - Receive NETMAIL from Unknown Domain, 1-2
  - Reference Materials, xvi
  - Remote Site Message, 3-10
  - Remote User Directory, 3-1
  - REMOTE USER DIRECTORY File (#, 2-1
  - REMOTE USER DIRECTORY File (#4.2997), 1-4
  - Revision History, iii
    - Documentation, iii
    - Patches, iv
  - Routines
    - ^XMRTCP, 7-2
    - XMLTCP, 12
    - XMRTCP
      - A, 2
      - XMRTCP, 11
  - Routing to Locally Known Ethernet Address, 7-1
- S**
  - SCP, 5-1
  - SCP Protocol, 2-4
  - SCP Protocols, 5-2
  - SCP), 2-4
  - Script Commands, 3-10

- Scripts
    - Big-Site Domain, 3-7
    - Playing, 4-3
    - Playing a Script, 3-7
    - Tips, 3-16
    - Transcripts, 3-7
  - Security
    - Transmission, 4-2
  - SEND Command, 3-11
  - Send Messages to Another UCI Option, 1-9
  - Sending and Receiving Messages Between UCIs, 9-1
    - Overview, 9-1
  - Sending and Receiving Network Messages, 3-1
  - Sending Network Messages, 3-3
  - Sequential Media Message Reception Option, 1-9
  - Sequential Media Queue Transmission Option, 1-10
  - Setting up a DECNET Channel for MailMan, 8-1
  - Setting up MailMan and TCP/IP, 3
  - Setting up More Domains to Use the TCP Channels
    - B, 2
  - Setting up Network MailMan
    - Protocols, 2-4
  - Setting up Network MailMan—Devices, 2-2
  - Show a queue Option, 1-8
  - Simple Communications Protocol, 2-4
  - Simple Mail Transport Protocol, 1-5
  - Simple Transport Protocol, 2-6
  - Simple Twisted Pair, 2-2
  - Site Parameters Option, 1-5, 1-8
  - Site User Directories, 1-1
  - Size Limit of Network Transmission, 1-4
  - Sliding Window Protocol, 2-4, 5-1
  - SMTP, 1-5, 2-6, 2-7, 3-2, 3-8, 3-10, 3-16, 4-3, 5-3, 7-1, 1, 2, 3, 10, 2
  - Standards Used by Network MailMan, 2-6
  - Starting the TCP/IP Poller, 7-2
  - Statistics
    - Message Delivery, 1-2
  - Status of Network Messages Immediately After Sending, 3-3
  - Steps in the Transmission of Network Messages, 3-2
  - STP, 2-2
  - Subroutine editor Option, 1-10
  - SWP, 2-4, 5-1
  - SWP Protocol, 2-4, 2-5
  - SWP Protocols, 5-2
  - Symbols
    - Found in the Documentation, xiii
  - Synchronous Communications Links, 2-2
  - Syntax
    - Domain Names, 4-1
  - System Managers, 1-5
- T**
- Tables and Figures, ix
  - TaskMan, 2-3, 4-3, 4, 11, 1, 2
  - Tasks
    - DECNET, 8-1
    - TCP Poller, 4
  - TCP/IP, 1-1, 1-2, 1-3, 1-5, 2-2, 2-5, 4-2, 5-1, 7-1, 1, 2, 3, 5, 10, 11, 1
    - Overview, 7-1
    - Poller, 7-2
  - TERMINAL TYPE File (#3.2, 2-1, 2-2, 2-3, 5-4, 3
  - Test Protocol, 2-6
  - Transcripts, 3-7
  - transmission failures, 4-4
  - Transmission Management Menu, 1-5, 1-8, 9-1
  - Transmission Process, 4-2
  - Transmission Queues, 3-6
  - Transmission Script, 6-1
  - TRANSMISSION SCRIPT Field, 6-1
  - TRANSMISSION SCRIPT File (#4.6, 2-1, 3-14, 3
  - Transmission Scripts, 1-2, 1-10, 3-2, 4-2, 6-1, 8-2, 9-1, 2
  - Transmission Scripts in the Domain File, 6-1
  - Transmission Security, 4-2
  - Transmit a Single Queue Option, 1-8
  - Transmit All Queues Option, 1-8
  - Transport Layer, 5-1
    - Overview, 5-1
  - Troubleshooting, 6-1
  - Trying Multiple Phone Numbers when one is Busy, 3-15
  - TURN command, 4-3, 1, 2
- U**
- URLs
    - Adobe Acrobat Quick Guide Web Address, xvi
    - Adobe Home Page Web Address, xvi
    - HSD&D Home Page Web Address, xvi
    - ISS
      - Acronyms Home Page Web Address, Glossary, 2
      - Glossary Home Page Web Address, Glossary, 2
      - MailMan Home Page Web Address, xvi
      - VistA Documentation Library (VDL) Home Page Web Address, xvi
  - Use this Manual, How to, xiii
  - Users Can Create their Own Network Mail Name, 1-3
  - Using MailMan and TCP/IP
    - A, 1
- V**
- Validation Number Edit Option, 1-10
  - VALIDATION NUMBER Field, 4-2
  - Variables
    - XMSTRIP, 3-15
  - Viewing a Network Message as a User, 3-3
  - VistA Documentation Library (VDL)
    - Home Page Web Address, xvi
  - VMS, 3, 4, 5, 11, 1
  - VMS Process Overview
    - A, 3
  - VOLUME SET(CPU) Field, 4-3
- W**
- WAIT Command, 3-11, 3-14
  - WAN, 1-10, 2-1, 2-5, 4-3, 5-1

## Index

### Web Pages

Adobe Acrobat Quick Guide Web Address, xvi  
Adobe Home Page Web Address, xvi  
HSD&D Home Page Web Address, xvi  
ISS  
    Acronyms Home Page Web Address, Glossary, 2  
    Glossary Home Page Web Address, Glossary, 2  
MailMan Home Page Web Address, xvi  
VistA Documentation Library (VDL) Home Page Web Address, xvi  
What is in the Queue?, 3-5  
Wide Area Network, 1-10, 2-1, 4-3, 5-1

## X

XECUTE Command, 3-11, 3-16  
XMCHRIS Option, 1-5, 1-6  
XMEDIT-DOMAIN-VALIDATION# Option, 1-10  
XMHIST Option, 1-7  
XMINET.EXE, 2, 3, 5, 6, 9, 10  
XMINET.INFO File  
    A, 2  
XMLIST Option, 1-7  
XMLTCP Routine, 12  
XMNET Menu, 1-4, 1-6  
    Functional Description, 1-4

XMNETE-TRANSMISSION-MANAGEMENT Menu, 1-8  
XMNETHELP Option, 1-6  
XMNET-QUEUE-MANAGEMENT Menu, 1-5, 1-6  
XMNET-TRANSMISSION-MANAGEMENT Menu, 9-1  
XMNET-TRANSMISSION-MANAGEMENT Menus, 1-5  
XMQACTIVE Option, 1-7  
XMQDISP Option, 1-7  
XMQDOMAINS Option, 1-6  
XMQSHOW Option, 1-8  
XMQUEUED Option, 1-7  
XMROB, 10-1  
    Overview, 10-1  
XMR-SEQ-RECEIVE Option, 1-9  
XMRTCP Routine, 11  
    A, 2  
XMR-UCI-RCV Option, 1-9  
XMR-UCI-SEND Option, 1-9  
XMSCRIPTEDIT Option, 1-9  
XMSCRIPTPLAY Option, 1-9  
XMSITE Option, 1-5, 1-8  
XMS-SEQ-TRASMITE Option, 1-10  
XMSTARTQUE Option, 1-8  
XMSTARTQUE-ALL Option, 1-8  
XMSTRIP Variable, 3-15  
XMSUBEDIT Option, 1-10