

# Introduction

**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 Decentralized Hospital Computer Program (DHCP) remain constant. Version 7.1 of MailMan is a great improvement over the original release, and it brings to the DHCP 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 7.1, is now in place. It 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 the VA Network MailMan package.**

**Version 7.1 of VA Electronic Mail System, known as MailMan, includes automatic interfaces with the VA' FileManager software through the use of bulletin cross-references and to parts of the VA operating system, named the Kernel. This version has been designed to be installed separately from the rest of the Kernel software.**

**MailMan introduces the new 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 it's functionalities.**

**Transmission Control Protocol/Internet Protocol (TCP/IP) channels will enable the use of sending/receiving mail between DHCP systems and non-DHCP systems (local and remote) running MailMan. This gives DHCP 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.**

## **Related Manuals**

**MailMan Version 7.1** documentation includes a **User Manual**, **Release Notes/Installation Guide**, a **Package Security Guide**, a **Programmer Reference Manual**, a **Technical Manual and Systems Management Guide**, and this **Network Reference Manual** along with its software. These manuals instruct on how to implement and maintain many special features using all components of MailMan. The **Technical Manual and Systems Management Guide**, and the **Package Security Guide** outline system management utilities and pertinent security requirements. The **Programmer Reference Manual** discusses new and existing **Application Programming Interface (API)** capabilities. The **VA Kernel** and **VA FileMan** documents must also be referenced.

## **How to Use this Manual**

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.

## **Menu Structure**

**XMNET** contains all options that are used for managing the network side of MailMan. Option descriptions for this menu are discussed in the **Network Management Menu** section of this document. **Queuing**, **queue displays**, **polling**, and **testing functions** may also be accessed through the use of these options.

## New 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 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 FTPd 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, if Subdomain of Known Domain**

In previous versions of MailMan, a domain entry needed to exist for each site that mail would be received from. In this version, you can receive mail from any subdomain of a domain in your Domain file.

**Example:** Previously, if you had AA.VA.GOV in your Domain file, you could not receive mail from SUB.AA.VA.GOV unless SUB.AA.VA.GOV was in your Domain file also. Now 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.

## **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, 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 subdomain of a particular domain.

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

MailMan V.7.1 is not case sensitive to e-mail addressing. It can handle lower case, upper case and mixed case (both lower and upper cases) of e-mail addresses. However, when sending an e-mail message out to a non-MailMan e-mail system (i.e., 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 may Create their Own Network Mail Name**

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

The Mail name is used as a return address when you send a message to someone whose mail box 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. 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.

## **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. Further discussion of the Remote User Directory may be found as part of the Appendix VII -- MailLink Program of the Technical Manual and 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 Kernel Site Parameters file. System managers can also control the maximum number of lines per message for transmission by editing Field #8.3 of the Kernel Site Parameters file to control the number of lines that may be transmitted by an interactive user without programmer access.

## Functional Description

**XMNET in the Options file is the Network Management menu (under the Manage MailMan [XMMGR] menu). The first options that must be used when starting Network MailMan are:**

- 1. Christen a Domain**
- 2. Site Parameters**

**While the local system requires some set up, the network side must be monitored. The options under XMNET 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 and 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, mail boxes, 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 mail box is located on another system.**

**System managers need to:**

- Provide support in updating the Domain file 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.**

## Network Management Menu

The **Network Management [XMNET]** menu allows the user to control the network editing, playing, and listing of scripts; managing domains; christening subordinate domains; and controlling pollers.

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

Use the **Christen a domain [XMCHRIS]** option 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.

The **Compare Domains in System Against Released List [XMQDOMAINS]** option lists domains released with the Kernel. It is in a table in routine XMYPOST that is run against the Domain file 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.

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

**Queue Management [XMNET-QUEUE-MANAGEMENT]** is a submenu of queue management options.

The **Actively Transmitting/Receiving Queues Report [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.

## Introduction

### Network Management Menu

The **Display message queues [XMQDISP]** option is used by the Postmaster to display the message queues of network sites.

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

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

The **Queues with Messages to Transmit Report [XMQUEUED]** option 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.

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

The **Transmit a Single Queue [XMSTARTQUE]** option 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 TaskManager job that will deliver the messages for the domain requested.

Use the **Transmit All Queues [XMSTARTQUE-ALL]** option if messages are queued in the Postmaster's baskets for a number of domains. TaskManager 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 TaskManager job, nor will it be requeued if it is empty or actively in the process of transmitting.

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

**Transmission Management [XMNETE-TRANSMISSION-MANAGEMENT]** is a submenu of transmission management options.

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

The **Play a script [XMSCRIPTPLAY]** option interprets the chosen script.

The **RECEIVE MESSAGES FROM OTHER UCI [XMR-UCI-RCV]** option is the corollary to another called Send Messages To Other 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.

The **SEND MESSAGES TO ANOTHER UCI [XMR-UCI-SEND]** option is the corollary to one called Receive Messages From Anther UCI [XMR-UCI-RCV]. Please see the above menu option for a complete description of both options and their usage.

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

The **Sequential Media Queue Transmission [XMS-SEQ-TRASMIT]** option 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 is not available. It can also be used for archiving.

Say a bulldozer knocked out your T1 line to the WAN in front of your computer room. It will be three weeks until the system is reconnected to the network. You know that a sister installation 10 miles down the road is still on line. You must get your payroll information to Austin. The messages are ready to be sent (in the queue to FOC-AUSTIN.VA.GOV). Quick ! Mount a tape. Use this procedure to "transmit" the queue onto the tape. Have the tape delivered to your

## **Introduction**

### **Network Management Menu**

**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.**

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

**The Validation Number Edit [XMEDIT-DOMAIN-VALIDATION#] option 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.**

# 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, the Remote User Directory, 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. The DHCP Tuning Guide should be read for further information. (Contact the Washington ISC if this publication is unavailable at your site.)

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
  - Domain - Transmission Script - Communications Protocol
  - Mail Box - Message Delivery Stats - InterUCI Transfer
  - Message Statistics - Kernel Site Parameters - Device
  - MailMan Time Zone - Network Senders Rejected
  - Terminal Type

## Setting up Network MailMan -- Devices

The devices that are used for sending and receiving Network Mail are generally released with the VA Kernel software. The document known as DHCP Tuning Guide describes how to set up an Integrated Data Communications Utility (IDCU) port and other devices. 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. Then there is an associated entry in the Terminal Type file. The example given below is used for modem connection and is not in the Tuning Guide:

```
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: $C(8)                               SUBTYPE: M-HAYESVAX
  TYPE: TERMINAL                                  TYPE-AHEAD: ALLOWED

NAME: M-HAYESVAX                                RIGHT MARGIN: 255
  FORM FEED: #                                    PAGE LENGTH: 66
  BACK SPACE: $C(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
```

### **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 if the capability is available. TCP/IP channels have been in use at 20 sites using VAX DSM for 6 months to a year as of June 1993, and the number increases steadily.

### **Notes on the Mini Out Device**

It appears that the new LatMaster software is 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 isn't 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 isn't. It will then time-out or get a data set hang up error. If there is no time-out 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.

### **For ALL Systems**

There are two other significant changes.

- Set the Open Time-out field in the Device file to seven.
- For active Network Mail ports, enter `S IO("C")=1,ZTNONEXT=1,%ZTIO=""` in the Close Execute field of the Terminal Type file.

This Close Execute property drops the current connection so that the device can be used to initiate a new one.

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



## **For VAX/DSM Sites**

### **History and Analysis**

The TaskMan submanagers of Kernel Version 6.5 actually closed the device they were using and re-opened it before trying to reuse it to do the next task that was queued. In Kernel Version 7, only the Close and Open executes were completed.

This meant that in Kernel 6.5, the connections that existed between sites were explicitly broken by the sender when the device was closed. Under Kernel 7, however, the connection could stay alive, because the device was never closed. On FORUM, the hundreds of tasks waiting for an IDCU device connection could fail repeatedly to make a new connection because the original IDCU connection was never dropped. Each would, in its turn, fail and then be rescheduled. Finally, when all had been processed this way, the submanager could close the device and make it available for the next task that tried to use it.

Each time a script was played, it would be successful, because a device had to be opened that had been previously closed. (New IDCU connections are newly opened, therefore, resetting ports always worked.) What was happening was determined by watching TaskMan using ZTMON.

IO queues were building up and going down, but the tasks would almost always be rescheduled. The failed transmission attempts generated messages and transcripts indicating that the new task was trying to talk to an IDCU logon, but was being asked for an ACCESS CODE. It was particularly probable at any site using Hunt Groups or Hunt Group devices.

In previous circumstances, it was bad for TaskMan to have MailMan close the port. In Version 6.5 and 7.0, the closes were slowly removed so that TaskMan would have full control. On VADATS this worked fine, since the close execute of the device brought down DTR and effectively recycled the port. The IDCU does not work this way.

Therefore, it is imperative that all VAX sites implement the following:

- Make sure that all incoming (IDCU) lines are tied to a LOGIN.COM file that sets it to a DIALIN type. The command SET TERM/DIAL should be in this file. This will cause any incoming network mail connection to be dropped when the device is closed and the job will be halted in routine ZU.

## 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.

<b>A UNIQUE NAME</b>	For example, SCP
<b>OPEN LOGIC</b>	Executable MUMPS code to be executed when the protocol is first initiated
<b>CLOSE LOGIC</b>	Executable MUMPS code executed when the protocol is ended
<b>READ LOGIC</b>	Executable MUMPS 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.
<b>SEND LOGIC</b>	Executable MUMPS code to send a line. Data to be sent is in XMSG and ER is set as with a READ.

### **Error Checking Communications Protocols** (SCP, ISCP, SWP, 3BSCP & TCP/IP-MAILMAN)

These protocols are fully checked procedures for reliable transmission over relatively noisy lines (switched modems). SCP is reliable, but at the expense of speed and efficiency, since every line is checked line-by-line. 3BSCP is an improvement that performs block-mode error checking. 3BSCP can speed up network transmissions when used, but is very sensitive to problems that may disrupt communications. 1SCP is more efficient than SCP. SWP 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. TCP/IP-MAILMAN is a protocol that 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) which protocol to use.

The first protocol to be used (SCP) was very slow (15-40 characters per second (cps)). 1SCP protocol was created out of SCP to put the hardest work into an operating system-dependent function and double transmission speeds. 1SCP 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 protocol 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.

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 3BSCP under ideal circumstances, running at 250 to 300 cps, but when circumstances are more adverse, it will succeed while 3BSCP may have multiple retries.

SWP protocol 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 protocol 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 protocol 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 more true 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 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**

This 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 Defense Advanced Research Program Agency (DARPA) and described in ARPA RFC 821, a Military Standard approved for use by the DoD (see 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. Discussion of on and an example of set up for VMS users so that a remote site can send images FTP to your site is located in the MailMan Version 7.1 Systems Management document.**

**Network Architecture**  
**Setting up Network MailMan -- Protocols**

# The 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 @, followed by their domain name.
- To send a message to CROMBIE at domain ANYWHERE-ELSE.VA.GOV, you would respond with, "CROMBIE@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:

**HALL,ROBERT W.@BIGSITE.VA.GOV will be changed to  
HALL.ROBERT\_W+@BIGSITE.VA.GOV**

## The Application Layer Overview

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 the 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.

## The Application Layer Steps in the Transmission of Network Messages

### Network Senders Rejected

The system manager can enter partial or full matches to network addresses into this file. 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 the Message

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 WILLIAMS,WILLIAM 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: <RET>
Send mail to: WILLIAMS,WILLIAM//<RET>
And sent to: WILL@ANYWHERE.VA.GOV      via ANYWHERE.VA.GOV
And send to: WILL@BIG-SITE.VA.GOV     via BIG-SITE.VA.GOV
And send to: <RET>

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

### Status of Message 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: WILLIAMS,WILLIAM in 'IN' basket.
Local Message-ID: 1842@ISC-ANY.VA.GOV

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

## Viewing a 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 @ symbol (i.e., From: <ABERCROMBIE@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.

**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:** 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:

- 1) If there are more than 10 recipients, MailMan will issue the message "too many recipients to list".
- 2) The message may be forwarded locally (or at another recipient's location), which would not show up here.

**In-reply-to:** If this message was entered as a reply to other messages, that message identifier is listed here.

**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.

## The Application Layer

### Steps in the Transmission of Network Messages

#### Example of Viewing a Message as a User

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 (query at the sending site)  
Subj: Test message 14 Jan 90 10:47 9 Lines  
From: WILLIAMS,WILLIAM in 'IN' basket.  
Local Message-ID: 1838@ISC-ANY.VA.GOV

WILLIAMS,WILLIAM Last read: 14 Jan 90 11:16  
WILL@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 (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: "WILLIAMS,WILLIAM"@@ISC-ANY,VA.GOV,  
WILL@ANYWHERE.VA.GOV,  
WILL@BIG-SITE.VA.GOV,  
WILL@MAR-AFB.AF.MIL

WILLIAMS,WILLIAM Last read: 14 Jan 90 11:21  
Select MESSAGE Action: IGNORE (in IN basket)// <RET> Ignored.

## What is in the Queue?

### Use the Queues with Messages to Transmit Report option.

QUEUES WITH MESSAGES TO GO OUT  
At FORUM.VA.GOV

26 Jun 93 10:49  
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

### 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		DANIELS,DANIEL (IRM - Albuq	
[7274361] 06/25/93 @ 15:56	2	NOT Selected [PROVIDENCE]	
2. Copy of: BAR CODING		ADAMS,ADAM (VAMC PROVID	
[7274466] 06/25/93 @ 16:08	388	NOT Selected [PROVIDENCE]	
3. 1358 DAILY RECORD		THOMAS,TOM E (VAMC LAKE	
[7274538] 06/25/93 @ 16:17	6	NOT Selected [PROVIDENCE]	
4. INTERIOR DESIGNER		WALTERS,WALT (VAMC LEXINGT	
[7274683] 06/25/93 @ 16:29	6	NOT Selected [PROVIDENCE]	
5. COPY OF: DATABASE		<"DAVIDS,DAVID_V+"@LIVERMORE.V	
[7275665] 21 JUN 93 11:50	54	NOT Selected [PROVIDENCE]	
*6. HIV/AIDS INFO		<MICHAELS,MIKE@SANFRANCISCO.VA	
[7275802] 25 JUN 93 16:15	3	NOT Selected [PROVIDENCE]	

**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 entry).**

## The Application Layer Steps in the Transmission of Network Messages

### Transmission Queues

This example shows that there is one message waiting 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
```

### An Example Showing Active Transmissions

```
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    _1LTA9802: TCP    8003322  110   1995001   1
900.5

Total Domains: 3, Total Messages Queued: 40
```

## Network Scripts

### Playing a Script & Script Transcripts

The example below 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. See the VA Kernel Site Parameters for details.

### Script for the Big-Site Domain

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

```
Script:
 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
```

### Transcript:

```
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
```

## The Application Layer Network Scripts

```
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: "WILLIAMS,WILLIAM"@ISC-ANY.VA.GOV,
11:14:09 S: WILL@BIG-SITE.VA.GOV,
11:14:13 S: WILL@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

### 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
```

### 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
```

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

## The Application Layer Network Scripts

### Remote Site Shows New Message

**The following message will appear at the remote domain.**

```
Subj: TEST MESSAGE 14 Jan 90 11:25 PDT 2 Lines
From:<WILLIAMS@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:<WILLIAMS@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: ,WILLIAMS@ISC-ANY.VA.GOV>
Subject: TEST MESSAGE
Date: 14 Jan 90 11:25 PDT
Message-ID:<1842@ISC-ANY.VA.GOV>
To: "WILLIAMS,WILLIAM"@ISC-ANY.VA.GOV,
```

```
WILL@BIG-SITE.VA.GOV
WILL@ANYWHERE.VA.GOV
WILLIAMS,WILLIAM Last read: 14 Jan 90 11:57
Select MESSAGE Action: IGNORE (in IN basket)// <RET>
```

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

## Network Script Commands

Application staff can set up scripts to interface to devices by writing scripts. These script commands may be used in conjunction with calls to the MailMan Script Processor. (See Appendix IX -- The MailMan Script Processor of the MailMan Programmer Reference Manual.) 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:

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

## **OPEN Command**

The format of the OPEN command is:

<u>Command Name</u>	<u>Command Description</u>
OPEN	H = [Domain] D = [Device] P = [Protocol]
Where	H = Specifies a queue to be sent D = Specifies a device to be used for sending P = Specifies a protocol to be used, which must be found in the Communications Protocol file

## **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 (See Queue Management).

## **LOOK Command**

There are two possible formats for this command, the old format and the new format introduced in Version 7.0. Both formats are valid under version 7.1.

### **LOOK - Original Format Form**

In the original format, the LOOK command is:

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

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
```

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

### **LOOK - New Format Form**

The new format for the LOOK command is: LOOK:T M{:L} {M{:L}}[TL] {M{:L}} where:

- "T" is the number of seconds until time-out (after which branch to line L2, if the third parameter is defined; else, abort the script).
- "M" is a 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" is the number of the line of the script to jump to on time-out.

There are several format rules under the old and new formats.

- For the new format, "|" must be the first character after the first space. Under the old format, "|" may not be the first character following the first space.
- Under the new format, 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.

## The Application Layer Network Script Commands

### Examples:

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

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

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|
```

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

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

**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|
```

This is a little tricky. The old syntax for looking for a string used SP(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 for.

Note also the difference between Examples 1 and 2 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:** The old and new formats are equally acceptable for performing the LOOK command.

## CALL Command

The format is:

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

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

**Example:** C MINIOUT calls a subroutine in File #4.6 to connect to a remote CPU.

## WAIT Command

The format of this command is:

```
WAIT [seconds]
```

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

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

## 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

The format of this command is:

```
DIAL NUM [num]
```

[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 variable XMSTRIP 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
```

## **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.

## **ERROR Command**

The format is:

```
ERROR [MSG]
```

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 has the format:

```
X [MUMPS code]
```

Any MUMPS code may be inserted. This command should be used with caution, as the full power of the MUMPS interpreter is available here. The transcript will read "executing \_\_\_\_\_".

## **MAIL Command**

The format of this command is MAIL. This command invokes the SMTP sender, and is generally the last command in the script. The script commands leading up to this 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 command 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
```

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

**The Application Layer  
Network Script Commands**

# The 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. Each label must begin with an alpha character and be between 1 and 12 characters long. Labels must contain only alphas, numerics, and the dash (-) character. Names may not end with a dash (-). Domain names are case-insensitive, upper and lower case alpha characters are treated the same. Names should be stored in the Domain file in all upper case format.

**Legal:**           ABC.DEF.GHI  
                  A123456789.B999999999  
                  TWO-SHOE.SHUFFLE

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

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 DHCP 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 DHCP sites.

### Domain Name Management

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

After a site has been assigned a name, they have administrative authority to create subdomains 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 subdomain 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 SUBDOMAINS. This is an administrative fact, not a judgment on how much power or primacy subdomains 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 @ in a MailMan address, the following procedures are followed:

1. The full domain name is looked up in the Domain file, using standard 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:

1. If the domain selected has a relay domain, that domain is selected. This allows the Domain file 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**

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. Please see the addendum on TCP/IP MailMan for further information on the subject.

## **Transmission Security**

MailMan's most recent security feature is a new field of the Domain file called Validation Number. This feature is used to make communications more secure via Network Mail transmission. Appendix XI -- MailMan Validation Numbers in the Technical Manual and Systems Management Guide describes uses of Validation Numbers.

## **Devices**

Upon installation of MailMan, the physical link devices (the outgoing communications lines) to Wide Area Network (WAN) or modems must be set up. These devices should belong to a Hunt Group. The TaskManager 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 (see Progress Report Logic), the TaskManager 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.

## **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.

## **The 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 Kernel Site Parameters file 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.**

then

- 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 \*.**

## **The Network Layer Overview**

# The 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 effected 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."

## Protocols

### **Communication Protocols** (SCP, 1SCP, SWP, 3BSCP)

MailMan currently supports four communication protocols: SCP and 1SCP for line-by-line checking, 3BSCP for block-mode checking, and Sliding Window Protocol (SWP). It is recommended that DHCP sites run 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**

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 Protocol**

SWP protocol 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).

## Sample Script Using SWP Protocol

```
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, MINIOUO Protocol SWP
Calling script 'MINIENGINE'
Waiting 4 seconds
S: A
Looking for 'name:'
R: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
R: J-2462-85-9494
R: Please Log in:
R: ~J
R: ~Juser name:
Xecuting W $(24,18,8,4), "VA688V: "_XMHOST_" ;password", $(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
```

## **The Transport Layer Protocols**

```
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
```

## 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 must point to the appropriate Terminal Type which names the specific modem attached to the port.**
- 2) For each type of modem attachable to the VA Kernel, a specific Terminal Type must be defined in the file for the modem. Modem Terminal Types begin with M-.**

**Modem Hang Up Logic must contain a valid MUMPS 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 MUMPS 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.**

## **The Transport Layer Modem Support**

# Network Mail Trouble Shooting

## 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 doesn't 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 subdomains. No protection exists to prevent infinite loops of transmissions that can occur when the parent has no entry for the subdomain and therefore implicitly assumes that it should route the message to the subdomain's creator.

### **Changes to DHCP Domains**

Until early 1991, domain names were used only by DHCP 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, DHCP domain names now will end with the ".MED.VA.GOV" suffix.

The heart of the Network Mail system of the Decentralized Hospital Computer System (DHCP) is the Domain file. 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 to all sites running DHCP 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 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 DHCP sites. This document applies only to those domains whose use is intended for all DHCP sites (nationally).**

**Information that is distributed with the Domain file must be kept current. All sites should have timely updates of changes to the Domain file 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 that should be coordinated include:**

- **A change to the MailMan Host field in the Domain file. 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
```

- **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
- 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.

***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. 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.

## **Network Mail Trouble Shooting Overview**

# 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 DHCP domain).**

**Here are some of the many possible suffixes in the Domain file:**

COM	EDU	MIL	ARPA	NET	O	RG	CA
UK	BITNET	UUCP	FI	DE	GOV	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: COM File that Adds Routine Info for IP Address**

```
#! 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 1.x.y.z with the appropriate X.25 IP address of
```

## Network Mail and TCP/IP Overview

```
$! 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 1.x.y.z 1 !to get to REMOTE SITE B
$
$! for SITE B
$! to enable routing, replace 1.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 1.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.
```

## Starting the TCP/IP Poller

**From a privileged account, start ^XMRTCP like this**

```
$!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
```

**Reminder! Also change the code at JOB^XMRTCP to the same. Keep a DCL context and you can then spawn out your FTP scripts, as well as a number of other things.**

# 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 Version 3.18 or later and the VA Kernel Version 6 or later 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**
- **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.

### Step 1

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.
```

### Step 2

Enter a transmission script in the Domain file 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 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, where xxxx=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
```

### Step 3

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 MUMPS tag^routine (DECNET^XMR). Change lower case 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/NOENABE DECNET^XMR
```

**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/NOENABE DECNET^XMR
```

### Step 4

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
```

### Step 5

On the remote host, create an account substituting your values for the following:

```
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-
```

## Network Mail Across a DECNET Channel

### Instructions for Completing a DECNET Channel

```
/password=PASS???.priv=(group,grpnam,tmpmbx,net,bx)-
/defpriv=(group,grpnam,tmpmbx,netmbx)/bytlm=100000/prclm=10/queprio=4
/NOPWDEXPIRED/PWDLIFE=NONE/NOACCESS/NETWORK
```

```
Username: NETMAILvol Owner: DSM
Account: xxxxxxx UIC: same group as DSM
DCL Tables: DCLTABLES
Default: DISK$whatever:[NETMAIL]
```

```
LGICMD: NETMAILvol_LOGIN
Login Flags:
```

```
Primary days: Mon Tues 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: 60 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 Priviledges:
GROUP GRPNAM TMPMBX NETMBX
Default Priviledges:
GROUP GRPNAM TMPMBX NETMBX
```

## Step 6

**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
```

## Network Mail Across a DECNET Channel Instructions for Completing a DECNET Channel

```
Object = NETMAIL vol

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

**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.**

### Step 7

**Create the following 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/NOENABLE DECNET^XMR
```

**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/NOENABLE DECNET^XMR
```

### Step 8

**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
```

**Network Mail Across a DECNET Channel  
Instructions for Completing a DECNET Channel**

# 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 (refer to the MailMan Technical Manual and 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. In the first step, messages in the queue are "dumped" to ^%ZISL. In the second step, the 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 (see option to Christen a Domain). The names of both UCIs (their domain names) must be entries in the Domain files 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 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" below.

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

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 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.

## **Sending and Receiving Messages Between UCIs Overview**

# New 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. (For example, ben.cs.umd.edu contains 4 dot-pieces.) This prevents XXX.YYY.MIL from being matched by 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 (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, the following Domain file entry will be automatically created for EEE.

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

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 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.

**New HELO Processing (XMROB)  
Overview**

## Appendix I -- MailMan Version 7.1 and TCP/IP

### Overview

**This section describes how to set up a DEC VAX system to be able to use MUMPS 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 7.1 (the MUMPS 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 DHCP 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. 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.

## The 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 MUMPS processes using TCP/IP channels, data is read and then parsed appropriately into segments. A special applications level protocol is necessary to do this.

### Receiver Flowchart

INET\_SERVERS => SERVERS.DAT => XMINET.EXE => XMINET.COM => MUMPS => MailMan

(The channel is passed to a MUMPS 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.

### Sender Flowchart (MUMPS Routines/Files)

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

- Scripts are in the Domain file.
- 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

- 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 MUMPS 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.

## MUMPS Process Overview

- Load NEW routines (XMRTCP & XMLTCP)
- Set up entries in:
  - 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) for that domain.
- Record the current Domain file 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
```

Stop the process named "INET\_SERVERS".

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

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

## Appendix I -- MailMan and TCP/IP

### Setting up MailMan and TCP/IP

=====  
**The following pages illustrate 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$: [XMINET]
LGICMD:  NL:
Flags:   DisCtly Restricted DisWelcome DisNewMail DisMail DisReport
Primary days:  Mon Tue Wed Thu Fri
Secondary days:                Sat Sun
Primary      000000000011111111112222  Secondary 000000000011111111112222
Day Hours    012345678901234567890123  Day Hours  012345678901234567890123
Network:     ##### Full access #####      ##### Full access #####
Batch:       ----- No access -----      ----- No access -----
Local:       ----- No access -----      ----- No access -----
Dialup:      ----- No access -----      ----- No access -----
Remote:      ----- No access -----      ----- No access -----
Expiration:  (none)                          Pwdminimum: 6  Login Fails: 0
Pwdlifetime: (none)                          Pwdchange: 13-MAR-1992 06:31
Last Login:  (none) (interactive), 13-MAR-1992 10:05
(non-interactive)
Maxjobs:     0  Fillm:      150  Bytlim:      40960
Maxacctjobs: 0  Shrfillm:   0  Pbytlim:      0
Maxdetach:   0  BIOlm:     18  JTquota:     1024
Prclm:       2  DIOLm:    18  WSdef:       900
Prio:        10 ASTlm:    24  WSquo:      1200
Queprio:     0  TQElm:    10  WSextent:   1500
CPU:         (none) Enqlm:  100  Pgflquo:   10240
Authorized Privileges:
  TMPMBX OPER NETMBX SHARE
Default Privileges:
  TMPMBX OPER NETMBX SHARE
Identifier                               Value                Attributes
DSM$MANAGER_DSMMGR                       %X80010004
```

**Warning!** 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.

=====  
**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);
```

## Appendix I -- MailMan and TCP/IP

### Setting up MailMan and TCP/IP

```
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    */
                    stsflg);    /* Net/Detach  */

    /* 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 */
}

```

## Appendix I -- MailMan and TCP/IP

### Setting up MailMan and TCP/IP

```
    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 */
}
}
```

## 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 mail box 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$trnlm("sys$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/envIRON=UUUUUUUU/uci=XXX/vol=###/data="'x'" SOC25^XMRTCP
$!-----
$ if debug then -      reply/user=UUUUUUUU "From the XMINET server -->OUT."
$ logout/brief.
```

**The two occurrences of UUUUUUUU should be changed to the username for the MUMPS 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.**

=====  
**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.

**Routine XMINET.INFO**

```
#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
#
#
```

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

First: Copy twg\$tcp:[netdist.etc]servers.dat servers\_wolly.dat

The above command 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 which 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. 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.

**Below is an example of what the file should look like with your change.**

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

## **MUMPS/VMS Software**

**There is one MUMPS routine that must ALWAYS BE RUNNING on your system on the node that the TCP/IP channel (X.25 board) is physically on.**

**This MUMPS 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. XMTCPCQUE.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.**

```
$! XMTCPCQUE.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
```

## **MUMPS Software**

### **Routines (in MailMan 7.1)**

#### **XMRTCP**

**This 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**

**This 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 File #3.4).**

## Other Mandatory Records

### Communications Protocol (3.4)

```
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.
```

### Device (3.5)

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

### Transmission Script (4.6)

```
NAME: TCPCHAN-SOCKET25
TEXT:
X S XMRPORT=25,X="ERRSCRPT^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")
```

### Domain (4.2)

```
NAME: FORUM.VA.GOV
MAILMAN HOST: 1.0.0.1
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
FLAGS: T
DISABLE TURN COMMAND: NO
```

### Terminal Type (3.2)

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

## Appendix II -- 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 are correct: `FLAGS`, `TRANSMISSION SCRIPT`.

However, in the usual DHCP 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 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 [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 Cmds, 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 for that domain.
- Record the current Domain file data for the domain you are about to change so it looks like the one below:

### Domain (4.2)

```
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
```

#### \*\*\*\* IMPORTANT! \*\*\*\*

- 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.

**Appendix II -- TCP/IP Poller**  
**Setting up More Domains to Use TCP Channels**

# 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 FileMan field.

**Checksum**

The result of a mathematical computation involving the individual characters of a routine or file.

**Cross-reference**

A cross-reference on a file provides direct access to the entries in several ways. For example, the Patient file is cross-referenced by name, social security number, and bed number. When asked for a patient, the user may then respond with either the patient's name, social security number, or bed number. Using cross-references speeds up access to the file for printing reports. A cross-reference is also referred to as an index or cross index.

**Data Dictionary**

A Data Dictionary (DD) contains the definitions of a file's elements (fields or data attributes); relationships to other files; and structure or design. Users generally review the definitions of a file's elements or data attributes; programmers review the definitions of a file's internal structure.

**Data Dictionary Access**

A user's authorization to write/update/edit the data definition for a computer file, also known as DD Access.

**Default**

A response the computer considers the most probable answer to the prompt being given. It is identified by double slash marks (//) immediately following it. This allows you the option of accepting the default answer or entering your own answer. To accept the default, you simply press the enter (or return) key. To change the default answer, type in your response.

## Glossary

### **Delete**

The key on your keyboard (may also be called **rubout** or **backspace** on some terminals) which allows you to delete individual characters working backwards by placing the **cursor** immediately after the last character of the string of characters you wish to delete. The @ sign (the "shift-2" key) may also be used to delete a file entry or data attribute value. The computer will ask "Are you sure you want to delete this entry?" to insure you do not delete an entry by mistake.

### **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).

### **DHCP**

The **Decentralized Hospital Computer Program** of the Veterans Health Administration (VHA), Department of Veterans Affairs (VA). DHCP software, developed by the VA, is used to support clinical and administrative functions at VA medical centers nationwide. It is written in MUMPS and, via the Kernel, will run on all major MUMPS implementations regardless of vendor. DHCP is composed of packages that conform with namespacing and other DHCP standards and conventions.

### **Domain**

A site for sending and receiving mail.

### **Enter**

Pressing the return or enter key tells the computer to execute your instruction or command or to store the information you just entered.

### **Free Text**

The use of any combination of numbers, letters, and symbols when entering data.

### **Help Prompt**

The brief help that is available at the field level when entering one or more question marks.

### **IDCU**

**Integrated Data Communications Utility**

### **Initialization**

The process of setting variables in a program to their starting value.

### **Input Transform**

An executable string of MUMPS code that is used to check the validity of input and converts it into an internal form for storage.

### **IRM**

**Information Resource Management**

**ISC**  
**Information Systems Center**

**Keyword**

A reference name that calls a help frame when entered at a message prompt.

**Kernel**

A set of DHCP software routines that function as an intermediary between the host operating system and the DHCP application packages such as Laboratory, Pharmacy, IFCAP, etc. The Kernel provides a standard and consistent user and programmer interface between application packages and the underlying MUMPS implementation. Two Kernel components, VA FileMan and MailMan, are self-contained to the extent that they may stand alone as verified packages. Some of the Kernel components are listed below along with their associated namespace assignments:

VA FileMan	DI
MailMan	XM
Sign-on/Security	XU
Menu Management	XQ
Tools	XT
Device Handling	ZIS
Task Management	ZTM

**Key**

A security code that is assigned to individual users that allows access to options.

**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.

## Glossary

### **Menu**

A list of options you are authorized access to and may select from.

### **Menu Tree**

A series of menus you sequence through in order to get to the specific option you desire.

### **Message-ID**

A message identifier that shows the time of transmission, the message number and the domain name of the message.

### **Modem**

A device for connecting a terminal to a telephone line, allowing it to communicate with another modem. Modems include the following types.

- Direct Connect*    The modem is directly hooked into the phone line.
- Acoustic*            The modem is connected to the telephone through the handset.
- Auto Answer*        When it detects a ring signal, the modem will "answer the phone."
- Auto Dial*            The modem, upon command from the terminal or the computer, will dial another modem.

### **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.

### **Namespacing**

A convention for naming DHCP package elements. The DBA assigns unique character strings for package developers to use in naming routines, options, and other package elements so that packages may coexist. The DBA also assigns a separate range of file numbers to each package.

### **On-Line**

A device is on-line when it is connected to the computer.

### **On-The-Fly**

A term given to the process of not permanently storing data in the data dictionary but having a computation performed at run time.

### **Password**

A user's secret sequence of keyboard characters, which must be entered at the beginning of each computer session to provide the user's identity.

**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.

**Pointer**

Points to another file where the computer stores information needed for the field of the file in which you are currently working. If you change any of the information in the field in which you are working, the new information is automatically entered into the "pointed to" file.

**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.

**Prompt**

The computer interacts with the user by issuing questions called **prompts**, to which the user issues a response.

**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.

**Read Access**

A user's authorization to read information stored in a computer file.

**Remote**

Any system that a user is not signed on to.

**Routine**

A program or a sequence of instructions called by a program, that may have some general or frequent use. MUMPS routines are groups of program lines that are saved, loaded, and called as a single unit via a specific name.

**SACC**

Standards and Conventions Committee of the Decentralized Hospital Computer Program.

## Glossary

### **Screen Editor**

This is VA FileMan's special screen-oriented text editor. This editor is used for the word-processing data type.

### **Script**

A set of MailMan commands and transmission scripts to a remote domain in the Domain file.

### **Server**

An automatic mail reader for internal messages.

### **System Manager/IRM Chief**

At each site, the individual who is responsible for managing computer systems, installing and maintaining new modules, and serving as liaison to the ISCs.

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

The primary transport protocol for MailMan

### **Surrogate**

A person who reads and/or sends mail in another user's name.

### **Transmission Script**

List of commands for directing a message stored in the transmission script field.

### **Trigger**

A trigger is an instruction that initiates a procedure. In VA FileMan, a trigger can be set up when entry of data in one field automatically updates a second field value.

### **Up-Arrow**

A character on your keyboard that looks like this: ^. The ^ character is used mainly for exiting or opting out of answering VA FileMan prompts and jumping to other fields in VA FileMan. The ^ character is the "shift-6" key on most keyboards.

### **User Access**

Access to a computer system. The user's access level determines the degree of computer use and the types of computer programs available. The systems manager assigns the user an access level. (See also **access code** and **programmer access code**.)

### **VA FileMan**

(also called VA FileManager)

A set of programs used to enter, maintain, access, and manipulate a database management system consisting of files. A package of on-line computer routines written in the MUMPS language that can be used as a stand-alone database system or as a set of application utilities. In either form, such routines can be used to define, enter, edit, and retrieve information from a set of computer-stored files.

**Validation Number**

A security check number that must be in the domains of both the sending and receiving sites.

**Verify Code**

An additional security precaution used in conjunction with the access code. Like the access code, it is also 6 to 20 characters in length and if entered incorrectly will not allow the user to access the computer. To protect the user, both codes are invisible on the terminal screen.

**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).

## **Glossary**

# Index

<b>3BSCP</b> .....	16, 17, 18, 47, 48
<b>Answer</b> .....	5
<b>Arriving Basket</b> .....	44
<b>Asynchronous Communications</b> .....	13
<b>Binary Large Objects</b> .....	1, 19
<b>BLOBs</b> .....	3, 19
<b>Block Mode</b> .....	16, 48
<b>Broadcast Messages</b> .....	45
<b>CALL Command</b> .....	33, 36
<b>Christen a Domain</b> .....	6, 7
<b>Communications Protocol</b> .....	82
<b>Compare Domains</b> .....	7
<b>DECNET</b> .....	59, 60, 61, 63
<b>Devices</b> .....	12, 43, 82
<b>DIAL Command</b> .....	33, 37, 38
<b>Display Message Queues</b> .....	8
<b>Domain File</b> .....	3, 4, 54, 65, 67
<b>Domain Name Server</b> .....	11
<b>Domain Names</b> .....	4, 21, 41, 42, 53
<b>Domains</b> .....	6, 11, 53, 82
<b>E Command</b> .....	29
<b>Edit a Script</b> .....	8
<b>Error Checking</b> .....	16
<b>ERROR Command</b> .....	33, 38
<b>Executable Image</b> .....	75
<b>File Transmission Protocol</b> .....	6
<b>FLUSH Command</b> .....	33, 37
<b>FTP</b> .....	6, 19
<b>HELO</b> .....	67
<b>Historical Queue</b> .....	8
<b>Hunt Group</b> .....	43
<b>IDCU</b> .....	12, 15, 29
<b>Images</b> .....	3
<b>Infinite Loops</b> .....	53
<b>Internet</b> .....	57
<b>Kernel Site Parameters File</b> .....	5, 44
<b>Kill Transmission</b> .....	45
<b>List a Transcript</b> .....	8
<b>LOOK Command</b> .....	33, 34, 35, 39
<b>MAIL Command</b> .....	33, 38
<b>Mail Name</b> .....	4
<b>Message Delivery Stats</b> .....	3
<b>Mini Out Device</b> .....	13
<b>Modem Support</b> .....	51
<b>MultiMedia Mail</b> .....	1
<b>Name Server</b> .....	53

## Index

Network Management Menu .....	6
Network Help .....	7
Network Management .....	7, 43
Network Senders Rejected .....	24
OPEN Command .....	33, 34
Playing a Script .....	8, 29, 44
Postmaster .....	27, 43, 44, 67
Progress Report Logic .....	44
Protocols .....	16
Query Display .....	31
Queue Management .....	6, 7, 43, 44
Queues .....	27, 44, 43
Queues Report .....	7
Queues with Messages .....	8
Remote User Directory .....	5, 11, 22
SCP .....	16, 17, 47, 48
Script Commands.....	33
Script Tips .....	39
Script Transcripts .....	29
SEND Command .....	33, 34
Send Messages to Another UCI .....	9
Sending and Receiving Network Messages .....	21
Sequential Media Queue .....	9
Show a Queue .....	8
Simple Communications Protocol .....	16
Simple Mail Transport Protocol .....	6
Simple Transport Protocol .....	18
Simple Twisted Pair .....	13
Site Parameters .....	6, 8
Site User Directories .....	1
Sliding Window Protocol .....	16, 47, 48
SMTP .....	6, 18, 19, 43, 57, 70, 85
Status of Message .....	24
STP .....	13
Subroutine Editor .....	10
SWP .....	16, 17, 18, 47, 48, 49
Synchronous Communications .....	13
System Managers .....	6
TaskMan .....	13, 15
TCP/IP .....	1, 3, 4, 6, 13, 16, 17, 18, 42, 47, 57, 58, 69, 70, 71, 73, 74, 80, 81, 83
Terminal Type .....	82
Transmission Failures .....	44
Transmission Management .....	6, 8, 65
Transmission Process .....	42
Transmission Queues .....	28
Transmission Script .....	53, 82
Transmit a Single Queue .....	8
Transmit All Queues .....	8
Trouble Shooting .....	53

<b>TURN Command</b> .....	<b>43, 70</b>
<b>Validation Number</b> .....	<b>10, 43</b>
<b>VAX/DSM Sites</b> .....	<b>15</b>
<b>Viewing a Message</b> .....	<b>25</b>
<b>VMS</b> .....	<b>71, 72, 73, 74, 81, 83</b>
<b>WAIT Command</b> .....	<b>33, 37</b>
<b>WAN</b> .....	<b>16, 17, 43, 47</b>
<b>XECUTE Command</b> .....	<b>33, 38</b>
<b>XMMGR</b> .....	<b>6</b>
<b>XMNET</b> .....	<b>6, 7</b>

# **Index**

U.S. Department of Veterans Affairs  
Decentralized Hospital Computer Program

**VA ELECTRONIC MAIL SYSTEM  
(MAILMAN)  
NETWORK REFERENCE MANUAL**

Version 7.1

June 1994

Information Systems Center  
Washington, DC



## Preface

**This manual will acquaint 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 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.**

## **Preface**

# Table of Contents

<b>Introduction .....</b>	<b>1</b>
<b>Related Manuals.....</b>	<b>2</b>
<b>How to Use this Manual.....</b>	<b>2</b>
<b>Menu Structure.....</b>	<b>2</b>
<b>New Network MailMan Features .....</b>	<b>3</b>
<b>MultiMedia Mail.....</b>	<b>3</b>
<b>Message Delivery Statistics.....</b>	<b>3</b>
<b>Network Delivery Script Error (E Command).....</b>	<b>3</b>
<b>Receive NETMAIL from Unknown Domain, if Subdomain of     Known Domain.....</b>	<b>3</b>
<b>Allow Site to Reject Network Mail from Particular Senders.....</b>	<b>4</b>
<b>Case Sensitive Message Fields from Network Receptions .....</b>	<b>4</b>
<b>Faster NETMAIL Transmission Speeds using TCP/IP     Channels.....</b>	<b>4</b>
<b>Users may Create their Own Network Mail Name.....</b>	<b>4</b>
<b>Forwarding Addresses Automatically Verified.....</b>	<b>5</b>
<b>Directory of Remote Recipients (MailLink) Expanded.....</b>	<b>5</b>
<b>Answering a Message Internet Style .....</b>	<b>5</b>
<b>Size Limit of Network Transmission.....</b>	<b>5</b>
<b>Functional Description.....</b>	<b>6</b>
<b>Network Architecture.....</b>	<b>11</b>
<b>Overview .....</b>	<b>11</b>
<b>Management Overview .....</b>	<b>11</b>
<b>Components of Network MailMan.....</b>	<b>11</b>
<b>Setting up Network MailMan -- Devices.....</b>	<b>12</b>
<b>Notes on the Mini Out Device .....</b>	<b>13</b>
<b>For VAX/DSM Sites.....</b>	<b>15</b>
<b>Setting up Network MailMan -- Protocols .....</b>	<b>16</b>
<b>Error Checking.....</b>	<b>16</b>
<b>Test Protocol .....</b>	<b>18</b>
<b>Standards used by Network MailMan .....</b>	<b>18</b>
<b>The Application Layer.....</b>	<b>21</b>
<b>Overview .....</b>	<b>21</b>
<b>Sending and Receiving Network Messages.....</b>	<b>21</b>
<b>Steps in the Transmission of Network Messages.....</b>	<b>23</b>
<b>Viewing a Message.....</b>	<b>25</b>
<b>Network Scripts.....</b>	<b>29</b>
<b>Playing a Script.....</b>	<b>29</b>
<b>Network Script Commands.....</b>	<b>33</b>
<b>OPEN Command.....</b>	<b>34</b>
<b>SEND Command.....</b>	<b>34</b>
<b>LOOK Command.....</b>	<b>34</b>
<b>CALL Command.....</b>	<b>36</b>
<b>WAIT Command.....</b>	<b>37</b>
<b>FLUSH Command.....</b>	<b>37</b>

## Table of Contents

<b>DIAL Command</b> .....	<b>37</b>
<b>ERROR Command</b> .....	<b>38</b>
<b>XECUTE Command</b> .....	<b>38</b>
<b>MAIL Command</b> .....	<b>38</b>
<b>The Network Layer</b> .....	<b>41</b>
<b>Overview</b> .....	<b>41</b>
<b>Syntax for Domain Names</b> .....	<b>41</b>
<b>Domain Name Management</b> .....	<b>41</b>
<b>Domain Name Resolution</b> .....	<b>42</b>
<b>Transmission Process</b> .....	<b>42</b>
<b>Transmission Security</b> .....	<b>43</b>
<b>Devices</b> .....	<b>43</b>
<b>Queues</b> .....	<b>43</b>
<b>Script Processing (Playing a Script</b> .....	<b>44</b>
<b>Progress Report Logic</b> .....	<b>44</b>
<b>The Postmaster</b> .....	<b>44</b>
<b>The Transport Layer</b> .....	<b>47</b>
<b>Overview</b> .....	<b>47</b>
<b>Protocols</b> .....	<b>48</b>
<b>Communication Protocols</b> .....	<b>48</b>
<b>SCP and 1SCP</b> .....	<b>48</b>
<b>Block Mode (3BSCP)</b> .....	<b>48</b>
<b>SWP Protocol</b> .....	<b>48</b>
<b>Modem Support</b> .....	<b>51</b>
<b>Network Mail Trouble Shooting</b> .....	<b>53</b>
<b>Overview</b> .....	<b>53</b>
<b>Transmission Scripts in the Domain File</b> .....	<b>53</b>
<b>Avoiding Infinite Transmission Loops</b> .....	<b>53</b>
<b>Changes to DHCP Domains</b> .....	<b>53</b>
<b>Network Mail and (TCP/IP</b> .....	<b>57</b>
<b>Overview</b> .....	<b>57</b>
<b>How to Route Internet</b> .....	<b>57</b>
<b>Network Mail Across a DECNET Channel</b> .....	<b>59</b>
<b>Overview</b> .....	<b>59</b>
<b>Setting up a DECNET Channel for MailMan</b> .....	<b>59</b>
<b>How to Use a DECNET Channel</b> .....	<b>59</b>
<b>Instructions for Completing a DECNET Channel</b> .....	<b>60</b>
<b>Sending and Receiving Messages Between UCIs</b> .....	<b>65</b>
<b>Overview</b> .....	<b>65</b>
<b>New HELO</b> .....	<b>67</b>
<b>Overview</b> .....	<b>67</b>
<b>Appendix I -- MailMan Version 7.1 and TCP/IP</b> .....	<b>69</b>

<b>Overview .....</b>	<b>69</b>
<b>Using MailMan and TCP/IP.....</b>	<b>70</b>
<b>Arriving.....</b>	<b>70</b>
<b>Sending.....</b>	<b>70</b>
<b>The TCP/IP Channel .....</b>	<b>71</b>
<b>VMS Process Overview.....</b>	<b>71</b>
<b>Setting up MailMan and TCP/IP.....</b>	<b>72</b>
<b>MUMPS Process Overview.....</b>	<b>72</b>
<b>Maintenance.....</b>	<b>73</b>
<b>Installation.....</b>	<b>73</b>
<b>Create and Install Executable Image.....</b>	<b>75</b>
<b>XMINET.COM.....</b>	<b>79</b>
<b>MUMPS/VMS Software.....</b>	<b>81</b>
<b>MUMPS Software.....</b>	<b>81</b>
<b>Other Mandatory Records.....</b>	<b>82</b>
<b>Appendix II -- TCP/IP Poller .....</b>	<b>83</b>
<b>Overview .....</b>	<b>83</b>
<b>Setting up More Domains to Use the TCP Channels.....</b>	<b>85</b>
<b>Glossary.....</b>	<b>87</b>
<b>Index.....</b>	<b>95</b>

## **Table of Contents**