



**MAILMAN
PROGRAMMER MANUAL**

Version 7.1 & Patch XM*7.1*50

July 1999

Department of Veterans Affairs
VISTA Software Development
Infrastructure Product Line

Preface

The "MailMan V. 7.1 & Patch XM*7.1*50 Programmer Manual" provides descriptive information and instructions on the use of the MailMan software within the VA's Veterans Health Information Systems and Technology Architecture (VISTA) environment. This document is intended for all personnel who use VISTA's MailMan software, and it emphasizes the changes made to the MailMan interface introduced by MailMan Patch XM*7.1*50 (i.e., Patch 50).

The MailMan Development Team appreciates and encourages all feedback regarding MailMan and its use. Please Use the National Online Information Sharing (NOIS) system to report any problems encountered and the Electronic Error and Enhancement Requests (E3R) system to request enhancements you would like incorporated into the MailMan software and documentation. Although we can't guarantee their incorporation, the MailMan Development Team will consider your suggestions for future enhancements to MailMan.

Table of Contents

Orientation	Orientation-1
Introduction	Introduction-1
PART I: Classic MailMan APIs (Pre-Patch 50)	I-1
1. Retired APIs	1-1
2. Programmer Calls.....	2-1
3. Utility Calls	3-1
4. Creating a Message	4-1
Common Variables.....	4-1
5. Creating/Sending/Forwarding a Message	5-1
6. Creating and Sending a PackMan Message	6-1
7. Creating and Sending a Reply or Answer	7-1
8. Creating and Sending a Bulletin.....	8-1
9. Address Lookup	9-1
10. Getting Information About a Message	10-1
11. Getting Text From a Message	11-1
12. Mailbox, Basket, and Message Activities	12-1
13. Server Message Activities	13-1
14. Mail Group Activities	14-1
PART II: New MailMan APIs (Post-Patch 50)	II-1
15. User Information—^XMVVITAE	15-1
16. Message Actions—^XMXAPI.....	16-1
Errors.....	16-1
Parameter Definitions.....	16-1
Message Action Building Blocks—^XMXAPI	16-15
17. Mailbox Actions—^XMXAPIB.....	17-1
Errors.....	17-1
18. Basket Actions—^XMXAPIB	18-1
Errors.....	18-1

Table of Contents

19. Mail Group Functions and Actions—^XMXAPIG.....	19-1
Errors.....	19-1
20. Interactive User Actions—^XM & ^XMXAPIU.....	20-1
^XM.....	20-1
^XMXAPIU.....	20-3
21. Message Editing—^XMXEDIT.....	21-1
Errors.....	21-1
22. Message Security, Permission, and Restriction Functions and Routines—^XMXSEC	22-1
Errors.....	22-1
^XMXSEC.....	22-1
^XMXSEC1.....	22-11
^XMXSEC2.....	22-15
23. Message and Mailbox Utility Functions and Routines—^XMXUTIL.....	23-1
24. Date and String Utility Functions and Routines—^XMXUTIL1.....	24-1
25. Message Information Functions and Routines—^XMXUTIL2 & ^XMXUTIL3.....	25-1
^XMXUTIL2.....	25-1
^XMXUTIL3.....	25-15
Glossary.....	Glossary-1
Appendix A—Message Server Protocol.....	Appendix A-1
Appendix B—Efficient Use of the API.....	Appendix B-1
Appendix C—Looking Up Messages.....	Appendix C-1
Appendix D—Setting Up Bulletins.....	Appendix D-1
Index.....	Index-1

Orientation

How to Use this Manual

Throughout this manual, advice and instructions are offered regarding the use of MailMan including the changes brought about by Patch 50 (i.e., XM*7.1*50) and the functionality it provides for Veterans Health Information Systems and Technology Architecture (VISTA) software products.

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

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

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

Symbol	Description
	Used to inform the reader of general information including references to additional reading material.
	Used to caution the reader to take special notice of critical information.

Table 1: Documentation Symbol Descriptions

- Descriptive text is presented in a proportional font (as represented by this font).
- "Snapshots" of computer online displays (i.e., roll-and-scroll screen captures/dialogues) and computer source code are shown in a *non*-proportional font and enclosed within a box.
 - User's responses to online prompts will be in boldface type.
 - The "<RET>" found within these snapshots indicate that the user should press the Enter or Return key on their keyboard.
 - Author's comments are displayed in italics or as "callout" boxes.



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

- All uppercase is reserved for the representation of M code, variable names, or the formal name of options, field and file names, and security keys (e.g., the XUPROGMODE key).

Assumptions About the Reader

This manual is written with the assumption that the reader is familiar with the **VISTA** computing environment.

It provides an overall explanation of MailMan Application Programmer Interfaces (APIs) and the changes contained in MailMan Patch 50 (i.e., XM*7.1*50). However, no attempt is made to explain how the overall **VISTA** programming system is integrated and maintained. Such methods and procedures are documented elsewhere. We suggest you look at the various VA home pages on the World Wide Web (WWW) for a general orientation to **VISTA**. For example, check out the following web sites:

- Veterans Health Information Systems and Technology Architecture (**VISTA**), formerly known as the Decentralized Hospital Computer Program (DHCP) System:
 - <http://www.va.gov/vama.htm#DHCP>
 - <http://vawww.va.gov/vama.htm#DHCP>
- **VISTA** Software Development Home Page: <http://vista.med.va.gov/>

Related Manuals and Other References

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

- "MailMan V. 7.1 & Patch XM*7.1*50 Release Notes"
- "MailMan V. 7.1 & Patch XM*7.1*50 Getting Started Manual"
- "MailMan V. 7.1 & Patch XM*7.1*50 User Manual"
- "MailMan V. 7.1 & Patch XM*7.1*50 Technical Manual"
- "MailMan V. 7.1 Network Reference Manual"
- MailMan Home Page at the following web address:

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

This site contains additional information and documentation.

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

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



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

<http://vista.med.va.gov/infrastructure/acrobat/index.html>



Introduction—Programmer APIs

<p>Topics To Be Discussed:</p>	<ul style="list-style-type: none">• Classic MailMan APIs (Pre-Patch 50)<ul style="list-style-type: none">➤ Retired APIs➤ Programmer Calls➤ Utility Calls➤ Creating a Message➤ Creating/Sending/Forwarding a Message➤ Creating and Sending a PackMan Message➤ Creating and Sending a Reply or Answer➤ Creating and Sending a Bulletin➤ Address Lookup➤ Getting Information About a Message➤ Getting Text From a Message➤ Mailbox, Basket, and Message Activities➤ Server Message Activities➤ Mail Group Activities
---------------------------------------	---



All *documented* Pre-Patch 50 APIs still work. Entry points *not* documented are subject to change and are *not* supported. Use them at your own risk!

Those *VISTA* applications *not* using approved MailMan APIs or having an approved Database Integration Agreement (DBIA) with MailMan, must review their code to determine if modifications are necessary *prior* to installing Patch 50.

Topics To Be Discussed (continued):

Topics To Be Discussed (continued):	<ul style="list-style-type: none">• New APIs (Post-Patch 50)<ul style="list-style-type: none">➤ User Information (^XMVVITAE)➤ Message Actions (^XMXAPIO)➤ Mailbox Actions (^XMXAPIOB)➤ Basket Actions (^XMXAPIOB)➤ Interactive User Actions (^XMXAPIOU)➤ Message Editing (^XMXAPIOEDIT)➤ Message Security, Permissions, and Restrictions (^XMXAPIOSEC, ^XMXAPIOSEC1, and ^XMXAPIOSEC2)➤ Message and Mailbox Utilities (^XMXAPIOUTIL)➤ Date and String Utilities (^XMXAPIOUTIL1)➤ Message Information Utilities (^XMXAPIOUTIL2 and XMXAPIOUTIL3)
--	---

MailMan has many documented programming entry points. In addition to the "classic" MailMan Application Programmer Interfaces (APIs), new APIs have been created with MailMan Patch 50 (i.e., XM*7.1*50). These new MailMan APIs, similar to VA FileMan APIs, were created to support other front ends (e.g., a Graphical User Interface [GUI]) to MailMan. Where possible, existing MailMan code has been altered to use the new APIs.

Often, MailMan entry points require the existence of certain variables when they are invoked. Some variables are understood to exist as they are set up during system security at signon through Kernel's software. These variables include: DUZ, DTIME, DT, U, IO, IOST, IOSL, and IOF. Normally, these variables should not be reset or KILLED. The variables listed are used in at least one MailMan entry point.

Pre-Patch 50 APIs

The following table shows a subset of the pre-Patch 50 APIs, along with any comparable (silent, except where noted) counterparts introduced with Patch 50:

Routine	Entry	DBIA	Description
XM		10064	Programmer entry point into MailMan.
XM	\$\$NU	10064	Tell user how many new messages he/she has. New: QMBOX^XMXAPIB, QBSKT^XMXAPIB, \$\$BNMSGCT^XMXUTIL, \$\$TNMSGCT^XMXUTIL, \$\$NEWS^XMXUTIL
XM	EN	10064	Option entry point into MailMan.
XM	KILL	10064	Kill MailMan variables.
XM	N1	10064	Create a mailbox. New: CRE8MBOX^XMXAPIB
XM	NEW	10064	Create a mailbox. New: CRE8MBOX^XMXAPIB
XMA	REC	1284	Interactive Manage Messages in MailMan. New: READ^XMXAPIU
XMA0	ENTPRT	1230	Print message (Interactive).
XMA0	HDR	1230	Print message (Headerless).
XMA0	PR2	1230	Print a message. New: PRTMSG^XMXAPI
XMA03	\$\$REN	1150	Resequence a basket. New: RSEQBSKT^XMXAPIB
XMA11	\$\$INFO		Interactive make a message "Information Only."

Table 2: Pre-Patch 50 APIs With Comparable New APIs Introduced With Patch 50

Table 2 (continued):

Routine	Entry	DBIA	Description
XMA11A	WRITE	1233	Interactive send or answer a message.
XMA1B	KL	10065	Delete a message from a basket.
XMA1B	KLQ	10065	Delete a message from a basket and put it in the "WASTE" basket. New: DELMSG^XMXAPI
XMA1B	S2	10065	Put a message in a basket. New: MOVEMSG^XMXAPI
XMA1C	REMSBMSG	10072	Remove a message from a Postmaster server basket. New: ZAPSERV^XMXAPI
XMA1C	SETSB	10072	Put a message into a Postmaster server basket. New: PUTSERV^XMXAPI
XMA2	GET	10066	Create a new message stub. New: CRE8XMZ^XMXAPI
XMA2	XMZ	10066	Create a new message stub. New: CRE8XMZ^XMXAPI
XMA21	CHK	10067	Check to see if the user is member of a group.
XMA21	DES	10067	Interactive addressing. Set the next default recipient. New: TOWHOM^XMXAPI (interactive)
XMA21	DEST	10067	Interactive addressing. Set the first default recipient. New: TOWHOM^XMXAPI (interactive)

Table 2: Pre-Patch 50 APIs With Comparable New APIs Introduced With Patch 50 (continued)

Table 2 (continued):

Routine	Entry	DBIA	Description
XMA21	INST	10067	Non-interactive remote addressing. New: TOWHOM^XMXAPI
XMA21	WHO	10067	Non-interactive addressing. New: TOWHOM^XMXAPI
XMA2R	\$\$ENT	1145	Reply to a message. New: REPLYMSG^XMXAPI
XMA2R	\$\$ENTA	1145	Answer a message. New: ANSRMSG^XMXAPI
XMAD2	\$\$BSKT	1147	Return the basket number after a lookup/create. New: CRE8BSKT^XMXAPIB, QBSKT^XMXAPIB
XMADGO	ZTSK	10068	Start tasks to deliver messages in the local delivery queues.
XMAH	ENT8	1040	Interactive list responses of a message.
XMAH1		1232	Interactive Reply to a message.
XMAH1	ENTA	1232	Interactive Reply to a message.
XMAREAD	\$\$CHECK	1149	NO LONGER SUPPORTED. Use ^DIR
XMAREAD	\$\$READ	1149	NO LONGER SUPPORTED. Use ^DIR
XMB		10069	Send a bulletin. New: TASKBULL^XMXAPI
XMB	BULL	10069	Interactive send a bulletin.
XMB	EN	10069	Send a bulletin, get a message number back. New: SENDBULL^XMXAPI

Table 2: Pre-Patch 50 APIs With Comparable New APIs Introduced With Patch 50 (continued)

Table 2 (continued):

Routine	Entry	DBIA	Description
XMBGRP	\$\$DM	1146	Delete members from a mail group.
XMBGRP	\$\$MG	1146	Create a new mail group or add members to an existing mail group.
XMCTLK	GO	1148	Interactive use device.
XMCU1	\$\$DECODEUP	1136	Convert ^ to ~U~ in a string (message subject). New: \$\$DECODEUP^XMXUTIL1
XMCU1	\$\$ENCODEUP	1136	Convert ~U~ to ^ in a string (message subject). New: \$\$ENCODEUP^XMXUTIL1
XMCU1	\$\$RTRAN	1136	Convert control characters to printable characters.
XMCU1	\$\$STRAN	1136	Convert printable characters back to control characters.
XMD		10070	Send a message. New: SENDMSG^XMXAPI
XMD	EN1	10070	Enter text in a message. If no recipients, prompt for them. Send the message.
XMD	ENL	10070	Add text to an existing message.
XMD	ENT	10070	Interactive send a message. New: SEND^XMXAPIU
XMD	ENT1	10070	Forward a message. New: FWDMSG^XMXAPI
XMD	ENT2	10070	Ask for additional recipients, then forward the message.
X MDF	\$\$ENT	1140	NO LONGER SUPPORTED. Use ENT1^XMD. NEW: FWDMSG^XMXAPI

Table 2: Pre-Patch 50 APIs With Comparable New APIs Introduced With Patch 50 (continued)

Table 2 (continued):

Routine	Entry	DBIA	Description
XMGAPI0	\$\$SUBCHK	1142	Validate the message subject. New: VSUBJ^XMXAPI
XMGAPI0	\$\$SUBGET	1142	Get the message subject. New: \$\$SUBJ^XMXUTIL2
XMGAPI1	\$\$EN	1048	Set up the MailMan environment for a user. New: INIT^XMVVITAE
XMGAPI1	\$\$READ	1048	Get the next line of message text.
XMGAPI2	\$\$HDR	1144	Get the message header information. New: INMSG^XMXUTIL2, INMSG1^XMXUTIL2, INMSG2^XMXUTIL2, INRESPS^XMXUTIL2
XMGAPI4	\$\$NU	1201	Get the new message information. (Private DBIA with Kernel) New: QMBOX^XMXAPI, \$\$NEWS^XMXUTIL
XML	GET	1283	Access message text as if by a server.
XMPG	ENT	10071	Create and send a PackMan message with globals.
XMRENT	\$\$NET	1143	Get message information.
XMS1	\$\$SRVTIME	1151	Set status for server recipient of a message.
XMS1	\$\$STATUS	1151	Get status of server recipient of a message.
XMS3	REC	10073	Get the next line of text from a message.
XMUT7	ENT	1132	Send a test message to a user's forwarding address.

Table 2: Pre-Patch 50 APIs With Comparable New APIs Introduced With Patch 50 (continued)

New APIs

The following is a table of the new APIs introduced in Patch 50, and any comparable pre-Patch 50 counterparts:

Routine	Entry	DBIA	Description
XM	CHECKIN	10064	Option ENTRY action.
XM	CHECKOUT	10064	Option EXIT action.
XM	HEADER	10064	Display user information/greeting when entering MailMan.
XMVVITAE	INIT	2728	Set up the MailMan user information. Original: \$\$EN^XMGAPI1
XMVVITAE	OTHER	2728	Set up the MailMan user environment when acting as a surrogate.
XMVVITAE	SELF	2728	Return to the MailMan user environment for self when finished acting as a surrogate.
XMXAPI	ADDRNSND	2729	Address and send a message. (Body already exists.)
XMXAPI	ANSRMSG	2729	Answer a message. Original: \$\$ENTA^XMA2R
XMXAPI	CRE8XMZ	2729	Create a new message stub. Original: GET^XMA2, XMZ^XMA2
XMXAPI	DELMSG	2729	Delete messages. Original: KLQ^XMA1B
XMXAPI	FLTRMSG	2729	Filter messages.
XMXAPI	FWDMSG	2729	Forward messages. Original: ENT1^XMD

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs

Table 3 (continued):

Routine	Entry	DBIA	Description
XMMAPI	LATERMSG	2729	Later messages.
XMMAPI	MOVEMSG	2729	Move messages to a basket. Original: S2^XMA1B
XMMAPI	PRTMSG	2729	Print messages. Original: PR2^XMA0
XMMAPI	PUTSERV	2729	Put a message in a server basket. Original: SETSB^XMA1C
XMMAPI	REPLYMSG	2729	Reply to a message. Original: \$\$ENT^XMA2R
XMMAPI	SENDERBULL	2729	Send a bulletin. Get back message number. Original: EN^XMB
XMMAPI	SENDMSG	2729	Send a message. Original: ^XMD
XMMAPI	TASKBULL	2729	Send a bulletin. Get back task number. Original: ^XMB
XMMAPI	TERMMSG	2729	Terminate messages.
XMMAPI	TOWHOM	2729	Non-interactive addressing. Original: INST^XMA21, WHO^XMA21
XMMAPI	VSUBJ	2729	Validate a subject. Original: \$\$SUBCHK^XMGAPI0

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XMXAPI	ZAPSERV	2729	Remove a message from the server basket. Original: REMSBMSG^XMA1C
XMXAPIB	CRE8BSKT	2723	Create a basket. Original: \$\$BSKT^XMAD2
XMXAPIB	CRE8MBOX	2723	Create a mailbox. Original: N1^XM, NEW^XM
XMXAPIB	DELBSKT	2723	Delete a basket.
XMXAPIB	FLTRBSKT	2723	Filter all messages in a basket.
XMXAPIB	FLTRMBOX	2723	Filter all messages in a mailbox.
XMXAPIB	LISTBSKT	2723	List baskets, optionally based on a partial match or new mail.
XMXAPIB	LISTMSGs	2723	List messages in one or all baskets, optionally based on the search criteria.
XMXAPIB	NAMEBSKT	2723	Change a basket's name.
XMXAPIB	QBSKT	2723	Get the basket information.
XMXAPIB	QMBOX	2723	Query a mailbox for new messages. Original: \$\$NU^XM, \$\$NU^XMGAPI4
XMXAPIB	RSEQBSKT	2723	Resequence a basket. Original: \$\$REN^XMA03
XMXAPIB	TERMMBOX	2723	Remove all traces of a user from MailMan.

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXAPIU	READ	2774	Interactive read/manage messages in a mailbox. Original: REC^XMA
XXAPIU	READNEW	2774	Interactive read new messages in a mailbox.
XXAPIU	SEND	2774	Interactive send a message. Original: ENT^XMD
XXAPIU	TOWHOM	2774	Interactive addressing. Original: DES^XMA21, DEST^XMA21
XXEDIT	CONFID	2730	Toggle the message's "Confidential" flag.
XXEDIT	CONFIRM	2730	Toggle the message's "Confirm Receipt Requested" flag.
XXEDIT	DELIVER	2730	Edit the message's delivery basket.
XXEDIT	INFO	2730	Toggle the message's "Information Only" flag.
XXEDIT	PRIORITY	2730	Toggle the message's "Priority" flag.
XXEDIT	SUBJ	2730	Edit the message's subject.
XXEDIT	TEXT	2730	Replace the message's text.
XXEDIT	VAPOR	2730	Edit the message's vaporize date.
XXSEC	\$\$ACCESS	2731	May this user access this message?
XXSEC	\$\$ANSWER	2731	May this user answer this message?
XXSEC	\$\$BCAST	2731	Was this message broadcast to all users?
XXSEC	\$\$CLOSED	2731	Is this message closed?
XXSEC	\$\$CONFID	2731	Is this message confidential?

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXMSEC	\$\$CONFIRM	2731	Is this message confirm receipt requested?
XXMSEC	\$\$COPY	2731	May this user copy this message?
XXMSEC	\$\$DELETE	2731	May this user delete/terminate this message?
XXMSEC	\$\$FORWARD	2731	May this user forward this message?
XXMSEC	\$\$INFO	2731	Is this message Information Only?
XXMSEC	\$\$LATER	2731	May this user "later" this message?
XXMSEC	\$\$MOVE	2731	May this user save/filter this message?
XXMSEC	\$\$ORIGIN8R	2731	Did this user originate (as sender or surrogate) this message?
XXMSEC	\$\$POSTPRIV	2731	Does this user have Postmaster privileges?
XXMSEC	\$\$PRIORITY	2731	Is this message priority?
XXMSEC	\$\$READ	2731	May this user read this message?
XXMSEC	\$\$REPLY	2731	May this user reply to this message?
XXMSEC	\$\$RPRIV	2731	Does this surrogate have READ privileges?
XXMSEC	\$\$RWPRIV	2731	Does this surrogate have READ/WRITE (send) privileges?
XXMSEC	\$\$SEND	2731	May this user send a message?
XXMSEC	\$\$SURRACC	2731	May this surrogate access this message?
XXMSEC	\$\$SURRCONF	2731	May this surrogate <i>not</i> read this message?
XXMSEC	\$\$WPRIV	2731	Does this surrogate have WRITE (send) privileges?
XXMSEC	\$\$ZCLOSED	2731	Is this message closed?
XXMSEC	\$\$ZCONFID	2731	Is this message confidential? (given XMZ)

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXMSEC	\$\$ZCONFIRM	2731	Is this message confirm receipt requested?
XXMSEC	\$\$ZINFO	2731	Is this message information only?
XXMSEC	\$\$ZORIGIN8	2731	Did this user originate (as sender or surrogate) this message?
XXMSEC	\$\$ZPOSTPRV	2731	Does this user have Postmaster privileges?
XXMSEC	\$\$ZPRI	2731	Is this message priority?
XXMSEC1	\$\$COPYAMT	2732	May the copy contain the requested responses?
XXMSEC1	\$\$COPYLIMS	2732	Returns a site's copy limits.
XXMSEC1	\$\$COPYRECP	2732	May the copy be sent to the original message's recipients?
XXMSEC1	\$\$PAKMAN	2732	Is this a PackMan message?
XXMSEC1	CHKLINES	2732	Checks whether a message is too long to be sent to a remote site.
XXMSEC1	CHKMSG	2732	Does this message exist and may the user access it?
XXMSEC1	GETRESTR	2732	Gets the restrictions on what a user can do with message.
XXMSEC1	OPTGRP	2732	What can the user do at the basket/message group level?
XXMSEC2	\$\$EDIT	2733	Can this user edit this message?
XXMSEC2	OPTEDIT	2733	If the user may edit the message, what exactly can be edited?
XXMSEC2	OPTMSG	2733	What may the user do with the message?

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXUTIL	\$\$BMSGCT	2734	Get the number of messages in a user's basket.
XXUTIL	\$\$BNMSGCT	2734	Get the number of new messages in a user's basket.
XXUTIL	\$\$BSKTNAME	2734	Get the basket name.
XXUTIL	\$\$NAME	2734	Get the user's name.
XXUTIL	\$\$NETNAME	2734	Get the user's network name.
XXUTIL	\$\$NEWS	2734	Get information on a user's new messages. Original: \$\$NU^XMGAPI4, \$\$NU^XM
XXUTIL	\$\$TMSGCT	2734	Get the number of messages in a user's mailbox.
XXUTIL	\$\$TNMSGCT	2734	Get the number of new messages in a user's mailbox.
XXUTIL	KVAPOR	2734	Edit the message vaporize date in a user's basket.
XXUTIL	LASTACC	2734	Record the last user access to this message.
XXUTIL	MAKENEW	2734	Make a message new.
XXUTIL	NONEW	2734	Make a message <i>not</i> new.
XXUTIL	PAGE	2734	Enter RETURN to continue or '^' to exit:
XXUTIL	WAIT	2734	Press RETURN to continue:
XXUTIL1	\$\$CONVERT	2735	Convert the Internet date/time to a VA FileMan date/time format.
XXUTIL1	\$\$CTRL	2735	Strip the control characters from a string.
XXUTIL1	\$\$DECODEUP	2735	Convert all ~U~ to ^ in a string. Original: \$\$DECODEUP^XMCU1

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXUTIL1	\$\$ENCODEUP	2735	Convert all ^ to ~U~ in a string. Original: \$\$ENCODEUP^XMCU1
XXUTIL1	\$\$GMTDIFF	2735	Get the time difference (+-hhmm) between a time zone and GMT.
XXUTIL1	\$\$INDT	2735	Convert VA FileMan date/time to an Internet date/time format.
XXUTIL1	\$\$MAXBLANK	2735	Reduce all three or more consecutive blanks in a string to two.
XXUTIL1	\$\$MELD	2735	Combine a string and a number to form a new string.
XXUTIL1	\$\$MMDT	2735	Convert VA FileMan date/time to MailMan date/time format.
XXUTIL1	\$\$SCRUB	2735	Strip the control characters and leading/trailing blanks from a string.
XXUTIL1	\$\$STRIP	2735	Strip leading/trailing blanks from a string.
XXUTIL1	\$\$TIMEDIFF	2735	Convert hour/hour fraction time difference to hour/minute time difference (+-hhmm).
XXUTIL1	\$\$TSTAMP	2735	Get a timestamp (\$H expressed in seconds).
XXUTIL1	\$\$ZONEDIFF	2735	Get the time difference (+-hhmm) from one time zone to the local one.
XXUTIL2	\$\$BSKT	2736	Which basket is a message in for a user?
XXUTIL2	\$\$DATE	2736	Get the message date.
XXUTIL2	\$\$FROM	2736	Get the message from.
XXUTIL2	\$\$KSEQN	2736	What is the sequence number of the message in this user's basket?

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXUTIL2	\$\$LINE	2736	How many lines are in the message?
XXUTIL2	\$\$NEW	2736	Is the message new for this user?
XXUTIL2	\$\$PRI	2736	Is the message priority?
XXUTIL2	\$\$QRESP	2736	Is this a response? Of which message? Which response number?
XXUTIL2	\$\$RESP	2736	How many responses does the message have?
XXUTIL2	\$\$SUBJ	2736	Get message subject.
XXUTIL2	\$\$ZDATE	2736	Get message date.
XXUTIL2	\$\$ZFROM	2736	Get message from.
XXUTIL2	\$\$ZNODE	2736	Get message zero node.
XXUTIL2	\$\$ZPRI	2736	Is message priority?
XXUTIL2	\$\$ZREAD	2736	How many responses has the user read?
XXUTIL2	\$\$ZSUBJ	2736	Get message subject.
XXUTIL2	INMSG	2736	Get the complete message information. Original: \$\$HDR^XMGAPI2
XXUTIL2	INMSG1	2736	Get the message information, Part 1. Original: \$\$HDR^XMGAPI2
XXUTIL2	INMSG2	2736	Get the message information, Part 2. Original: \$\$HDR^XMGAPI2
XXUTIL2	INRESP	2736	Get the response information. Original: \$\$HDR^XMGAPI2

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

Table 3 (continued):

Routine	Entry	DBIA	Description
XXUTIL2	INRESPS	2736	Get the message response count and responses read information. Original: \$\$HDR^XMGAPI2
XXUTIL3	Q	2737	List/Find the message addressees.
XXUTIL3	QD	2737	List/Find the message recipients.
XXUTIL3	QL	2737	List/Find the "latered" message addressees.
XXUTIL3	QN	2737	Get the network header lines.

Table 3: New APIs Introduced With Patch 50 With Comparable Pre-Patch 50 APIs (continued)

PART I: Classic MailMan APIs (Pre-Patch 50)

1. Retired APIs

The following APIs have been retired (superceded by others) and are no longer supported:

- `$$CHECK^XMAREAD` NO LONGER SUPPORTED. Use `^DIR` instead.
- `$$READ^XMAREAD` NO LONGER SUPPORTED. Use `^DIR` instead.
- `$$ENT^XMDF` NO LONGER SUPPORTED. Use `ENT1^XMD`.
- `$$EN^XMGAPII` NO LONGER SUPPORTED. Use `INIT^XMVVITAE`.

2. Programmer Calls

- **^XM**

Programmer entry point into MailMan. Meant to be used by a programmer to enter MailMan without going through the menu system. Gives a programmer access to many MailMan options.



This is a self-contained entry point, and it needs no other calls.

Input Variables:

DUZ User DUZ.

Output Variables:

None.

- **GO^XMCTLK**

This routine lets you interactively use a device and displays keyboard entry and data coming down the line. It is good for testing devices, network outgoing points, etc. The **VISTA** programming environment is assumed (initialized through **D ^XUP** or signon through **^XUS**). All I/O from the keyboard and device chosen are echoed on the screen. What is displayed on the screen may be captured into a mail message. Type an "A" to communicate with TalkMan.

3. Utility Calls

- **KILL^XM**

KILL any MailMan variables that may be left over from previous calls.

Input Variables:

None.

Output Variables:

None.

- **N1^XM**

Create a mailbox for a user.



Compare to NEW^XM described below and CRE8MBOX^XM APIB described in Chapter 17 in this manual.

Input Variables:

XMDUZ User's DUZ.

XMZ (optional) If you wish to prevent the user from seeing messages created before a certain date, then set XMZ to a message number in the MESSAGE file (#3.9). The user will not be able to access any messages created earlier than this one, unless the message is already in the user's mailbox or is forwarded to the user. This really only applies to users who left the organization and then returned, or if (heaven forbid) you are re-using a DUZ. This prevents the user from accessing old messages that may have been addressed to the user.

Output Variables:

None.

- **NEW^XM**

Create a mailbox for a user.



Compare to NI^XM described above and CRE8MBOX^XM APIB described in Chapter 17 in this manual.

Input Variables:

Y User's DUZ.

XMZ (optional) If you wish to prevent the user from seeing messages created before a certain date, then set XMZ to a message number in the MESSAGE file (#3.9). The user will not be able to access any messages created earlier than this one, unless the message is already in the user's mailbox or is forwarded to the user. This really only applies to users who left the organization and then returned, or if (heaven forbid) you are re-using a DUZ. This prevents the user from accessing old messages that may have been addressed to the user.

Output Variables:

None.

- **ZTSK^XMADGO**

Start tasks to deliver messages in local delivery queues.

Input Variables:

None.

Output Variables:

None.

- **\$\$DECODEUP^XMCU1(STRING)**

Takes a string, converts any ~U~ to ^, and returns the result.



This API is identical to \$\$DECODEUP^XMUTIL1 described in Chapter 1 in this manual.

Input Parameters:

STRING Any character string.

- **\$\$ENCODEUP^XMCU1(STRING)**

Takes a string, converts any ^ to ~U~, and returns the result.



This API is identical to \$\$ENCODEUP^XMXUTIL1 described in Chapter 1 in this manual.

Input Parameters:

STRING Any character string.

- **\$\$RTRAN^XMCU1(STRING)**

Takes a string that had been converted by \$\$STRAN^XMCU1 (described below), undoes the conversion, and returns the result.

Input Parameters:

STRING Any character string converted by \$\$STRAN^XMCU1.

- **\$\$STRAN^XMCU1(STRING)**

Takes a string, converts any control characters to printable characters, and returns the result. The conversion may be undone by \$\$RTRAN^XMCU1 (described above).

Input Parameters:

STRING Any character string.

- **ENT^XMUT7(Y)**

Send a test message to a user's forwarding address. The message is also sent to the Postmaster. If the forwarding address is no good, the Postmaster receives an error message.

Input Parameters:

Y DUZ of user, whose forwarding address you want to test.

4. Creating a Message

Common Variables

DUZ	User's DUZ. If DUZ is not defined, it defaults to the Postmaster. This is who is really sending the message.
XMDUZ	User's DUZ or FREE TEXT. This is from whom the message will appear to be. If it is not defined, it defaults to DUZ. (If DUZ is not defined, it defaults to Postmaster.) If it is FREE TEXT, it must not be more than 70 characters.
XMSUB	Subject of the message. It should be from 3 to 65 characters in length. If it is less than 3 characters, then three dots ("...") will be appended to it. If it is more than 65 characters, then it will be truncated.
XMTEXT	The name of the array (in open format) containing the text of the message. The array itself may be a local or a global variable, and it must be in a format acceptable to VA FileMan WORD-PROCESSING fields.
XMY	Addressee array, XMY(x)="" , where x may be: <ul style="list-style-type: none">• User's DUZ or enough of the user's name for a positive ID. For example: <code>XMY(1301)=""</code> OR <code>XMY("lastname,first")=""</code>• If the user or SHARED,MAIL is an addressee, the basket to place the message in may be specified. For example: <code>XMY(duz,0)=basket name or ien</code>• If SHARED,MAIL is an addressee, the automatic delete date of the message may be specified. For example: <code>XMY(.6,"D")=delete date in any format that VA FileMan understands.</code>• G.group name (enough for positive ID). For example: <code>XMY("G.group name")=""</code>

- **S.server name** (enough for positive ID).
- **D.device name** (enough for positive ID).
- Prefix the above (except devices and servers) by:

I: for "Information Only" recipient (may not reply). For example:

```
XMY("I:1301")="" or
XMY("I:lastname,first")=""
```

C: for "Copy" recipient (not expected to reply). For example:

```
XMY("C:1301")="" or
XMY("C:lastname,first")=""
```

L@datetime: for when (in future) to send to this recipient (datetime can be anything accepted by VA FileMan). For example:

```
XMY("L@25 DEC@0500:1301")="" or
XMY("L@1 JAN:lastname,first")="" or
XMY("L@2981225.05:1301")=""
```

(May combine **IL@datetime:** or **CL@datetime:**)

- To delete any recipient (including users, groups, devices, and servers, prefix with a hyphen/dash ("-"). For example:


```
XMY(-1301)="" or
XMY("-lastname,first")=""
```
- To address any recipient (including users, groups, devices, and servers) at a remote site, just add the @site name. For example:


```
XMY(recipient@site name)=""
```

- **\$\$INFO^XMA11(XMZ)**

Interactive, lets the user edit message XMZ's "Information Only" field and returns 0.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **XMZ^XMA2**

Create a new message stub in the MESSAGE file (#3.9). A message stub is an entry in the MESSAGE file with no text or recipients. The SUBJECT field (#.01) will be set to XMSUB. The FROM field (#1) will be set to XMDUZ. The SENT DATE/TIME field (#1.4) will be set to the current date/time, in internal VA FileMan format. If XMDUZ is a number and it differs from DUZ, then the SENDER field (#1.1) will be set to DUZ. If **XMDUZ=.5** and **DUZ'=.5**, the INFORMATION ONLY? field (#1.97) will be set to "y", thus, making the stub "Information only."



Compare to CRE8XMZ^XMXAPI described in Chapter 16 in this manual.

Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list above.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list above.

If it is zero, null, or not defined, it defaults to DUZ.

XMSUB For a description of this variable, please refer to the "Common Variables" list above.

Output Variables:

XMZ =Message number in the MESSAGE global, if stub creation succeeds.

=-1, if stub creation fails (if, for example, a lock on the MESSAGE global can't be achieved).

Example:

```
S XMSUB="TEST RESULTS",XMDUZ="TESTING PACKAGE" D XMZ^XMA2
```

This creates a message stub from the TESTING PACKAGE.

Once you have created a message stub and any time before you send the message, you may set other message type fields, by using VA FileMan as follows:

```
S DIE=3.9,DA=XMZ,DR="<field #>////<value>" D ^DIE
```

<u>Field #</u>	<u>Value</u>	<u>Causes message to be</u>
1.7	P	Priority
1.95	y	Closed
1.96	y	Confidential
1.97	y	Information only

For example, to force message 100213 to be priority and closed:

```
S DIE=3.9,DA=100213,DR="1.7////P;1.95////Y" D ^DIE
```

See ENL^XMD to find out how to add text to your message stub.

See ENT1^XMD to find out how to forward your message to various recipients.

See EN1^XMD to find out how to add text to your message stub and send your message to various recipients.

- **GET^XMA2**

Create a new message stub in the MESSAGE file (#3.9).



This API works exactly the same way as XMZ^XMA2 (described above), except that if the stub creation fails, your process will HALT! Thus, this API should *not* be used. Instead, use XMZ^XMA2 or CRE8XMZ^XMXAPI.

Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list above.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list above.

If it is zero or null or not defined, it defaults to DUZ.

XMSUB For a description of this variable, please refer to the "Common Variables" list above.

Output Variables:

XMZ =Message number in the MESSAGE global, if stub creation succeeds.

- **\$\$\$SUBCHK^XMGAPI0(XMSUB,XMFLG)**

Checks a message subject for validity and returns valid subject or error string explaining why it's not valid. Leading and trailing blanks are automatically removed. Up-arrows ("^") are automatically converted to ~U~.



Compare to VSUBJ^XM API described in Chapter 16 in this manual.

Input Parameters:

XMSUB	Message subject.
XMFLG	Interactive?
	0—no
	1—yes

Possible results, in actual order:

Subject is too long

Non-interactive: 3-Entered subject too long...^\$E(<subject>,1,65)

Interactive: "Entered subject too long..."

1^\$E(<subject>,1,250)

At this point, leading and trailing blanks are removed, and up-arrows ("^") are converted to ~U~.

Subject contains control characters

Non-interactive: 5-Subject cannot contain control characters.^<subject>

Interactive: "Control characters removed (<subject> is Subject accepted)." (Control characters are removed and checking continues.)

Subject is null

Non-interactive: <subject>

Interactive: <subject>

Subject is "?"

Non-interactive: 4-Enter a Message Subject, between 3 & 65 characters long or '^' to exit.^<subject>

Interactive: "Enter a Message Subject, between 3 & 65 characters long or '^' to exit."

1^<subject>

Subject is too short

Non-interactive: 1-SUBJECT must be at least 3 characters long.^<subject>

Interactive: "SUBJECT must be at least 3 characters long."

1^<subject>

Subject is reserved format

Non-interactive: 2-Subject names of this format (1""R""1.N) are RESERVED ^<subject>

Interactive: "Subject names of this format (1""R""1.N) are RESERVED"

1^<subject>

Subject is OK

Non-interactive: ^<subject>

Interactive: ^<subject>

5. Creating/Sending/Forwarding a Message

- **WRITE^XMA11A**

Interactive send a message. (It is the same as XMSEND, the Send a Message option.) It is similar to ENT^XMD (described below), but it does *not* display the MailMan greeting.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) User's DUZ.

Output Variables:

None.

- **^XMD**

Create and send message. If no recipients are defined, and '\$D(ZTQUEUED)', then prompt for them. Addressing restrictions are waived. (It's as if you set XMDF.)



Compare to SENDMSG^XMXAPI described in Chapter 16 in this manual.

Core Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMSUB For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMTEXT For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMY (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

If '\$D(XMY)<10' (no recipients), and '\$D(ZTQUEUED)' (job running in the foreground), the user will be prompted for recipients. If there are no recipients, the message will be created, but it won't be sent, and XMMG will not be defined.

Additional Input Variables:

- XMMG** (optional) If there are no recipients in XMY and the job is running in the foreground, XMMG may contain the default recipient presented to the user. If XMMG is not defined, then the default recipient is the user.
- XMSTRIP** (optional) String containing characters that should be removed from the message text. Default is none.
- XMROU** (optional) Array of routines to be loaded in a PackMan message. For each routine, set XMROU(x)="", where x is the routine name.
- DIFROM** (optional) Specifically for the VA FileMan package.
- XMYBLOB** (optional) Specifically for the Imaging package.

Output Variables:

- XMZ** =Message number in the MESSAGE global, if successful.
=unchanged or undefined, if failure.
- XMMG** This is the variable that the calling program should check to determine whether or not the call was successful.
=undefined, if successful
=String containing error message, if failure

Variables KILLED Upon Exit:

If the call is successful, the following variables are KILLED: XMSUB, XMTEXT, XMY, XMSTRIP, XMMG, and XMYBLOB.

If the call fails, those variables may or may not be KILLED, except for XMMG, which will contain an error string.

- **EN1^XMD**

Add text to a message, and address it and send it. If no recipients are defined, and '\$D(ZTQUEUED)', then prompt for them.



There is no other equivalent API.

Core Input Variables:

DUZ	For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMTEXT	For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMDF	(optional) If \$D(XMDF) all addressing restrictions are waived.
XMY	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual. If \$D(XMY)<10 (no recipients), and '\$D(ZTQUEUED)' (job running in the foreground), the user will be prompted for recipients. If there are no recipients, the message will be created, but it won't be sent, and XMMG will not be defined.
XMZ	Message IEN in the MESSAGE file (#3.9).

Additional Input Variables:

XMMG	(optional) If there are no recipients in XMY and the job is running in the foreground, XMMG may contain the default recipient presented to the user. If XMMG is not defined, then the default recipient is the user.
XMSTRIP	(optional) String containing characters that should be removed from the message text. Default is none.
XMROU	(optional) Array of routines to be loaded in a PackMan message. For each routine, set XMROU(x)="", where x is the routine name. (To create and send a PackMan message with globals in it, see ENT^XMPG.)
DIFROM	(optional) Specifically for the VA FileMan package.

Output Variables:

None.

Variables Killed Upon Exit:

XMTEXT, XMY, XMSTRIP, XMMG

- **ENL^XMD**

Add text to a message.



There is no other equivalent API.

Core Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMTEXT For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMZ Message IEN in the MESSAGE file (#3.9).

Additional Input Variables:

XMSTRIP (optional) String containing characters that should be removed from the message text. Default is none.

Output Variables:

None.

Variables Killed Upon Exit:

XMSTRIP

- **ENT^XMD**

Interactive send a message. (It is the same as XMSSEND, the Send a Message option.) This call may be placed in a menu option as follows:

Entry action: S XMMENU(0)=<name of the menu option>

Routine: ENT^XMD

Exit action: K XMMENU D CHECKOUT^XM



Compare to SENDMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

Output Variables:

None.

- **ENT1^XMD**

Forward a message. Addressing restrictions are waived. (It's as if you set XMDF.)



Compare to FWDMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMY For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

Variables Killed Upon Exit:

XMDUZ, XMY

- **ENT2^XMD**

Forward a message. If '\$D(ZTQUEUED)', prompt for additional recipients, whether or not any are already defined.



There is no other equivalent API.

Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDF (optional) If \$D(XMDF) all addressing restrictions are waived.

XMY (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

If '\$D(ZTQUEUED) (job running in the foreground), the user will be prompted for additional recipients.

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

Variables Killed Upon Exit:

XMDUZ, XMY

6. Creating and Sending a PackMan Message

- **ENT^XMPG**

Create and send a PackMan message with globals in it. If no recipients are defined, the message will be created, but it won't be sent anywhere. Addressing restrictions are waived. (It's as if you set XMDF.)



There is no other equivalent API.



To create and send a PackMan message with routines in it, use ^XMD described in Chapter 5 in this manual.

Core Input Variables:

DUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMDUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMSUB	For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMTEXT	String of open global roots, separated by semicolons. The globals are loaded into the PackMan message.
XMY	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
^TMP("XMP", \$J)	(optional) Text to be placed in the PackMan message. Must be in the format ^TMP("XMP", \$J, i, 0) = <text>

Additional Input Variables:

DIFROM	(optional) Specifically for the VA FileMan or KIDS package.
--------	---

Output Variables:

XMZ	=Message number in the MESSAGE global, if successful. =unchanged or undefined, if failure.
XMMG	=unchanged or undefined, if successful =String containing error message, if failure

Variables Killed Upon Exit:

XMY, ^TMP("XMP",\$J)

7. Creating and Sending a Reply or Answer

- **WRITE^XMA11A**

Interactive answer a message.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) User's DUZ.

X Must be set to "A", otherwise this call sends a new message.

XMZ Message IEN in the MESSAGE file (#3.9), of the message to which you are sending an answer.

Output Variables:

None.

- **\$\$SENT^XMA2R(XMZ,XMSUB,,XMREPLY,XMSTRIP,XMDUZ,XMNET)**

Creates and sends a reply to a message and returns the message number of the reply. If the reply is not successful, returns a string with the text of the error. Unlike an answer, a reply is sent to all (local) recipients of the message to which you are replying.



Compare to \$\$ENTA^XMA2R described below and ANSRMSG^XMXAPI and REPLYMSG^XMXAPI described in Chapter 16 in this manual.

Core Input Parameters:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

.XMREPLY Text of the reply. Must be in a local array passed by reference. It must be in a format acceptable to VA FileMan WORD-PROCESSING fields.

XMZ Message IEN in the MESSAGE file (#3.9), of the message to which you are replying.

Additional Input Parameters:

- XMNET** (optional) If the sender of the original message is at a remote site, should the reply be sent to the sender, too? (Ignored, unless message is from a remote sender.)
=0—no (default)
=1—yes
- XMSTRIP** (optional) String containing characters that should be removed from the reply text. Default is none.
- XMSUB** Subject of the reply (ignored, unless message is from a remote sender).

- **\$\$ENTA^XMA2R(XMZ,XMSUB,XMTEXT,XMSTRIP,XMDUZ)**

Creates and sends an answer to a message and returns the message number of the answer. If the answer is not successful, it returns a string with the text of the error. Unlike a reply, an answer is sent only to the sender of the original message. It makes no difference whether the sender is local or remote.



Compare to \$\$ENT^XMA2R described above and ANSRMSG^XMXAPI and REPLYMSG^XMXAPI described in Chapter 16 in this manual.

Core Input Parameters:

- DUZ** (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- XMDUZ** (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- XMSUB** Subject of the answer.
- XMTEXT** Text of the answer. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- XMZ** Message IEN in the MESSAGE file (#3.9), of the message to which you are sending an answer.

Additional Input Parameters:

- XMSTRIP** (optional) String containing characters that should be removed from the answer text. Default is none.

- **^XMAH1**

Interactive create and send a reply to a message.



There is no other equivalent API.

Core Input Variables:

DUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMDUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMK	IEN of the user's basket in which the message resides.
XMZ	Message IEN in the MESSAGE file (#3.9), of the message to which you are replying.

Additional Input Variables:

XMDF	(optional) If \$D(XMDF) all addressing restrictions are waived.
------	---

Output Variables:

None.

- **ENTA^XMAH1**

Interactive create and send a reply to a message. Identical to ^XMAH1.



For the description of this API, please refer to ^XMAH1 described above.

8. Creating and Sending a Bulletin

- **^XMB**

Create and send bulletin in the background. Unlike EN^XMB and SENDBULL^XMXAPI, the message number (XMZ) is not returned. The bulletin is sent to the mail groups defined for the bulletin in the BULLETIN file (#3.6), as well as to any additional recipients defined in XMY.



Compare to EN^XMB described below and SENDBULL^XMXAPI and TASKBULL^XMXAPI described in Chapter 16 in this manual.

Core Input Variables:

DUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMB	Full, exact name, of the bulletin. Case is important.
XMB(#)	(optional) Bulletin parameter(s). For example: <code>XMB(1)=<parm 1>, XMB(2)=<parm 2>, etc.</code>
XMDUZ	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

Additional Input Variables:

XMBTMP	(optional) If \$D(XMBTMP) do not initialize (KILL) the ^TMP addressee global, because it contains bulletin addressees.
XMDF	(optional) If \$D(XMDF) all addressing restrictions are waived.
XMDT	(optional) Date/time (in any format understood by VA FileMan) to send the bulletin. Default is now.
XMTEXT	(optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual. This is text, in addition to the text defined in the bulletin, to append to the bulletin.
XMY	(optional) Recipients, in addition to those defined in the bulletin. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

If the bulletin has no recipients and there are no recipients in XMY, the bulletin won't be sent, and there won't be any error indication.

XMYBLOB (optional) Specifically for the Imaging package.

Output Variables:

XMB =unchanged, if bulletin is found.
=-1, if bulletin is not found in BULLETIN file (#3.6).

Variables Killed Upon Exit:

XMTEXT, XMY

- **BULL^XMB**

Interactive create and send bulletin. This is the same entry point called by the XMPOST option. This is a good way to test a bulletin.



There is no other equivalent API.

Core Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

Output Variables:

None.

- **EN^XMB**

Create and send bulletin in the foreground. Unlike ^XMB and TASKBULL^XMXAPI, the message number (XMZ) is returned. The bulletin is sent to the mail groups defined for the bulletin in the BULLETIN file (#3.6), as well as to any additional recipients defined in XMY.



Compare to ^XMB described above and SENDBULL^XMXAPI and TASKBULL^XMXAPI described in Chapter 16 in this manual.

Core Input Variables:

DUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMB Full, exact name, of the bulletin. Case is important.

XMB(#) (optional) Bulletin parameter(s). For example:
 XMB(1)=<parm 1>, XMB(2)=<parm 2>, etc.

XMDUZ (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

Additional Input Variables:

XMBTMP (optional) If \$D(XMBTMP) do not initialize (KILL) the ^TMP addressee global, because it contains bulletin addressees.

XMDF (optional) If \$D(XMDF) all addressing restrictions are waived.

XMTEXT (optional) For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

This is text, in addition to the text defined in the bulletin, to append to the bulletin.

XMY (optional) Recipients, in addition to those defined in the bulletin. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

If the bulletin has no recipients and there are no recipients in XMY, the bulletin won't be sent, and there won't be any error indication.

XMYBLOB (optional) Specifically for Imaging package.

Output Variables:

XMZ =Message number in the MESSAGE global, if successful.
 =unchanged or -1, if failure.

XMB =undefined, if bulletin is found.
 =-1, if bulletin is not found in BULLETIN file (#3.6).

Variables Killed Upon Exit:

XMB, XMTEXT, XMY

9. Address Lookup

- **DES^XMA21**

Interactive addressing, with the next default recipient set. Unlike DEST^XMA21, XMY is not KILLED upon entry.



Compare to DEST^XMA21 described below and TOWHOM^XMXAPIU described in Chapter 1 in this manual.

Input Variables:

XMDUZ	DUZ of the user doing the addressing.
XMDF	(optional) If \$D(XMDF) all addressing restrictions are waived.
XMMG	(optional) The DUZ or name of the next default recipient.

Output Variables:

XMY	Array of addressees. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
XMOUT	If \$D(XMOUT), then the user aborted addressing.
X	= "^" if the user aborted addressing. = "" if finished addressing normally.

- **DEST^XMA21**

Interactive addressing, with the first default recipient set. Unlike DES^XMA21, XMY is KILLED upon entry.



Compare to DES^XMA21 described above and TOWHOM^XMXAPIU described in Chapter 1 in this manual.

Input Variables:

XMDUZ	DUZ of the user doing the addressing.
XMDF	(optional) If \$D(XMDF) all addressing restrictions are waived.
XMDUN	The name of the first default recipient.

Output Variables:

- XMY Array of addressees. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- XMOUT If \$D(XMOUT), then the user aborted addressing.
- X ="^" if the user aborted addressing.
 ="" if finished addressing normally.

- **INST^XMA21**

Non-interactive addressing. XMY is not KILLed upon entry.



Compare to DES^XMA21 described above and TOWHOM^XMXAPIU described in Chapter 1 in this manual.

Input Variables:

- XMDUZ DUZ of the user doing the addressing.
- X A local or remote address. (-X will remove any address.)
- XMDF (optional) If \$D(XMDF) all addressing restrictions are waived.
- XMLOC (optional) If \$D(XMLOC) then any error (in XMMG) will be displayed.

Output Variables:

- XMY Array containing the address. For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- Y Outcome of the address lookup.
 =<DUZ^full name>, if local address.
 =<domain ien^domain name>, if remote address.
 =-1, if the lookup failed (bad address in X).
- XMMG =error message, if Y=-1.
 ="via <domain name>", if remote address.
 ="" (null), otherwise.

- **WHO^XMA21**

Non-interactive addressing. This call is identical to INST^XMA21.



For the description of this API, please refer to INST^XMA21 described above.

10. Getting Information About a Message

- **ENT8^XMAH**

Interactive display a list of all responses to a message.

Input Variables:

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **\$\$SUBGET^XMGAPI0(XMZ)**

Returns the subject of a message. Any ~U~ are automatically converted to up-arrow ("^"). If message does not exist, returns null.



Compare to \$\$SUBJ^XMXUTIL2 and \$\$ZSUBJ^XMXUTIL2 described in Chapter 1 in this manual.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$HDR^XMGAPI2(XMZ,,ARRAY,FLAG)**

Sets up (in ARRAY) an array of information about a message. Returns one of the following:

- If successful, it returns zero ("0").
- If not successful, then one of the following is returned:
 - "1-Undefined message number"
 - "1-No message number"
 - "1-No such message"
 - "2-User is not a sender of recipient."
 - "4-Invalid user"



Compare to \$\$NET^XMRENT described below and INMSG^XMXUTIL2 and all other APIs in ^XMXUTIL2 described in Chapter 1 in this manual.

Input Variables:

- DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
- XMDUZ (optional) DUZ of the user. Default is DUZ.

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- FLAG Flag, which determines what message information is placed in the output ARRAY parameter.
- 0 or undefined—return basic information.
- 1—return basic information + response and BLOB count information.
- 91—return flag 1 information + response IDs.
- 92—return flag 1 information + BLOB IDs.
- 93—return all of the above.

Output Parameters:

- .ARRAY Message information array.
- If FLAG=0 or undefined:
- ("BROADCAST") = 1 if the message was broadcast; 0 otherwise.
- ("BSKT") Basket name (XMDUZ); null if not in basket.
- ("BSKT IEN") Basket IEN (XMDUZ); null if not in basket.
- ("DATE") Message date/time, in format MAY 25, 1999@08:16:00, if local, or as sent, if remote.
- ("DATE FM") Message date/time (VA FileMan format); date only if remote.
- ("LINES") Number of lines in the original message.
- ("NEW") = 1 if the message is new; 0 otherwise.
- ("PXMZ") Message number of original message; null if not a response.
- ("SENDER") Sender (from, external format).
- ("SENDER DUZ") Sender (from) DUZ; null if remote or fictitious.
- ("SUBJ") Subject (external format).
- ("SURROG") Surrogate sender (external format).
- ("TYPE") Message type (piece 7 of the message's zero node).
- ("XMZ") Message IEN in the MESSAGE file (#3.9).

1 = returns function value and value array above, also additional value array as follows:

("RRED")	Responses read (XMDUZ); null if not applicable.
("RRCV")	Responses received; null if not applicable.
("BLOBCNT")	Number of non-textual body parts attached (for the Imaging package).

If FLAG=91 and if the message has responses, returns value array as with Flag 1 and an array of response nodes and values as follows:

("RSP",i)	(Pointer to the MESSAGE file [#3.9]) array of responses.
-----------	--

If FLAG=92 and if the message has BLOBs, it returns the value array as with Flag 1 and an array of non-textual body parts as follows:

("BLOB",i)	(Pointer to the OBJECT file [#2005]) array of BLOBS (for the Imaging package)
------------	---

If FLAG=93, it returns all of the above.

- **\$\$NET^XMRENT(XMZ)**

Returns an ^-delimited string of information about a message. If message does not exist, returns null.

Returns:

- Piece 1: Message date, in the following format: MAY 25, 1999@08:16:00, if local, or as sent, if remote.
- Piece 2: Scramble hint, if any; otherwise null.
- Piece 3: Message from (external).
- Piece 4: Message ID at originating site (XMZ@sitename, if local).
- Piece 5: Message sender, usually surrogate (external).
- Piece 6: Message subject (external).
- Piece 7: Message ID of original message, if this is a reply (XMZ@sitename, if local).
- Piece 8: Message type (Piece 7 of the message's zero node).



Compare to \$\$HDR^XMGAPI2 described above and INMSG^XMXUTIL2 and all other APIs in ^XMXUTIL2 described in Chapter 1 in this manual.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

11. Getting Text From a Message

If you want the whole text of a message, use the VA FileMan function \$\$GET1^DIQ. To get a line at a time, use one of the following:

- **\$\$READ^XMGAPI1()**

Returns a line of text from a message. By calling this API repeatedly, you can retrieve the lines of text, in order, from start to finish. The only thing this function does is:

```
D REC^XMS3 Q XMRG
```

Thus, it's just another way of invoking REC^XMS3.



Compare to GET^XML and REC^XMS3 described below. For a description of the input and output variables, please refer to REC^XMS3.

Example:

```
S XMZ=message number in the MESSAGE file
F S LINE=$$READ^XMGAPI1() Q:XMER=-1 D
. ; line is in LINE, and also in XMRG
```

- **GET^XML**

Set up an executable variable (XMREC), which, when executed, retrieves the next line of text from a message. Thus, by executing XMREC repeatedly, you can retrieve the lines of text, in order, from start to finish. This call creates a lot of variables, some of which are documented here, just so you know what you're getting into.



Compare to \$\$READ^XMGAPI1 described above and REC^XMS3 described below.

Input Variables:

XMCHAN Must be set to "SERVER", or another protocol defined in the COMMUNICATIONS PROTOCOL file (#3.4).

Output Variables:

XMCHAN IEN, in the COMMUNICATIONS PROTOCOL file (#3.4), of the Input XMCHAN.

XMCLOSE Ignore this. It's the variable, which, when executed, closes the communications channel.

XMOPEN Ignore this. It's the variable, which, when executed, opens the communications channel.

XMPROT	Copy of Input XMCHAN.
XMREC	This is what we're after. It's the variable, which, when executed, retrieves the next line of text of a message.
XMSEN	Ignore this. It's the variable, which, when executed, sends the next line of text of a message.

Example:

```
S XMCHAN="SERVER" D GET^XML
```

Now, we've got, among other variables, XMREC, which, when executed, does, guess what? **D REC^XMS3**

Thus, it's just another way of invoking REC^XMS3. Continuing:

```
S XMZ=message number in the MESSAGE file
F X XMREC Q:XMER=-1 D
. ; line is in XMRG
```



For a description of the input and output variables, please refer to REC^XMS3 described below.

- **REC^XMS3**

Get a line of text from a message. By calling this API repeatedly, you can retrieve the lines of text, in order, from start to finish.



Compare to \$\$READ^XMGAPI1 and GET^XML described above.

Input Variables:

XMZ	Message IEN in the MESSAGE file (#3.9).
XMPOS	(optional) Line number of previous line read. Default is .99 . Thus, the first time this API is called, XMPOS should not be defined, unless you want to start reading the message at some line other than line 1. If it's a message from a remote site, and you want to read the header records, set XMPOS=0. For every subsequent call to this API, XMPOS is already set to read the next line, so you don't need to set it.

Output Variables:

XMER	End of text reached? 0—no -1—yes
XMPOS	Line number of the latest line read. (null, if XMER=-1)
XMRG	Text of latest line read. (null, if XMER=-1)

Example:

```
S XMZ=message number in the MESSAGE file
F D REC^XMS3 Q:XMER=-1 D
. ; line is in XMRG
```


12. Mailbox, Basket, and Message Activities

- **\$\$NU^XM(FORCE)**

Returns the number of new messages the user has, and it may display a message to the user telling how many.



Compare to QMBOX^XM APIB described in Chapter 17, QBSKT^XM APIB described in Chapter 1, \$\$BNMSGCT^XM UTIL, \$\$TNMSGCT^XM UTIL, and \$\$NEWS^XM UTIL described in Chapter 23.

Input Variables:

DUZ DUZ of user.

XMDUZ (optional) DUZ of user. Default is DUZ.

Input Parameters:

FORCE Force MailMan to display a message to the user, telling how many new messages the user has? "You have <x> new messages."

0—No, only display it if the user has received new messages since the last time the user was told.

1—Yes, display it.

- **REC^XMA**

Interactive read/manage messages. (It is the same as XMREAD, the Read/Manage Messages option.) This call may be placed in a menu option as follows:

Entry action: S XMMENU(0)=<name of the menu option>

Routine: REC^XMA

Exit action: K XMMENU D CHECKOUT^XM



Compare to READ^XM APIU described in Chapter 1 in this manual.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

Output Variables:

None.

- **ENTPRT^XMA0**

Interactive Print a message.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

XMDUZ (optional) DUZ of the user. Default is DUZ.

XMK Basket IEN.

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **HDR^XMA0**

Headerless Print a message.



Compare to PR2^XMA0 described below and PRTMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

DUZ For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.

IO Device to which to print.

XMDUZ (optional) DUZ of the user. Default is DUZ.

XMK Basket IEN.

XMTYPE (optional) If not defined, the message is printed in its entirety. Otherwise, ";"-piece 6 is the number of the response from which to print. (If ";"-piece 6 is null or zero, then the message is printed in its entirety.) If XMTYPE="^", then this API aborts.

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **PR2^XMA0**

Print a message.



Compare to HDR^XMA0 described above and PRTMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

DUZ	For a description of this variable, please refer to the "Common Variables" list in Chapter 4 in this manual.
IO	Device to which to print.
XMDUZ	(optional) DUZ of the user. Default is DUZ.
XMK	Basket IEN.
XMTYPE	(optional) If not defined, the message is printed in its entirety. Otherwise, ";"-piece 6 is the number of the response from which to print. (If ";"-piece 6 is null or zero, then the message is printed in its entirety.) If XMTYPE="^", then this API aborts.
XMZ	Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **\$\$REN^XMA03(XMDUZ, XMK)**

Performs an integrity check on a user's basket, resequences messages in a user's basket, and returns a string, "Resequenced from 1 to **n**", where **n** is the number of messages in the basket.



Compare to RSEQBSKT^XMXAPIB described in Chapter 1 in this manual.

Input Variables:

XMDUZ	DUZ of user.
XMK	Basket IEN.

- **KL^XMA1B**

Delete a message from a basket. Unlike KLQ^XMA1B, the message is *not* put in the "WASTE" basket.



Compare to KLQ^XMA1B described below and DELMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

XMDUZ	DUZ of the user who owns the basket.
XMK	(optional) Basket IEN. If '\$G(XMK), then MailMan will look to see which basket the message is in.
XMZ	Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **KLQ^XMA1B**

Delete a message from a basket and put it in the "WASTE" basket. Unlike KL^XMA1B, the message is put in the "WASTE" basket.



Compare to KL^XMA1B described above and DELMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

XMDUZ	DUZ of the user who owns the basket.
XMK	(optional) Basket IEN. If '\$G(XMK), then MailMan will look to see which basket the message is in.
XMZ	Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

- **S2^XMA1B**

Put a message in a basket. Be careful with this call, because it does not check to see if the message is already in another basket—it just puts it where you tell it, so you could end up with the same message in more than one basket.



Compare to MOVEMSG^XMXAPI described in Chapter 16 in this manual.

Input Variables:

XMDUZ	DUZ of the user who owns the basket.
XMKM	Basket IEN.
XMZ	Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

Variables Killed Upon Exit:

XMKM

- **\$\$BSKT^XMAD2(XMKN,XMDUZ)**

Given a basket name and a user's DUZ, look up the basket. If it doesn't exist, create it and return its IEN. If it does exist, return its IEN. If there's an error, return an error message.

Input Variables:

XMDUZ	DUZ of user.
XMKN	Basket name.

13. Server Message Activities

- **REMSBMSG^XMA1C**

Delete a message from a server basket. The message is *not* put in the "WASTE" basket. All server baskets belong to the Postmaster, so it is not necessary to specify which user. Packages should call this API when they have processed the server message, otherwise it will never be removed from the basket. The MailMan purge routines do not touch messages in server baskets.



Compare to ZAPSERV^XMXAPI described in Chapter 16 in this manual.

Input Variables:

XMSER Server name. Must be the full name, starting with "S."

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

Variables Killed Upon Exit:

XMKD, XMZ, XMDUZ, XMK, XMSER

- **SETSB^XMA1C**

Put a message in a server basket. All server baskets belong to the Postmaster, so it is not necessary to specify which user. Generally, packages will not be using this call, because MailMan delivers server messages to server baskets.



Compare to PUTSERV^XMXAPI described in Chapter 16 in this manual.

Input Variables:

XMXX Server name. Must be the full name, starting with "S."

XMZ Message IEN in the MESSAGE file (#3.9).

Output Variables:

None.

Variables Killed Upon Exit:

None.

- **\$\$SRVTIME^XMS1(XMZ,XMSER,STATUS)**

Sets the LAST READ DATE/TIME field (#2) and STATUS field (#5) of a (server) message recipient in the RECIPIENT multiple of a message in the MESSAGE file (#3.9). (There is no other equivalent API.) Field #2 is set to the current date/time. Field #5 is set to the STATUS parameter. Returns 0, if no error; otherwise returns one of the following:

"1 No Update"—If XMSER not found in the RECIPIENT multiple.

"2 Status too long"—If STATUS is longer than 30 characters.

"3 Bad Characters in Status"—If STATUS contains an up-arrow ("^").

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMSER Server name. Must be the full name, starting with "S."

STATUS Status string to put in the STATUS field.

- **\$\$STATUS^XMS1(XMZ,XMSER)**

Returns the STATUS field (#5) of a (server) recipient from the RECIPIENT multiple of a message in the MESSAGE file (#3.9). (There is no other equivalent API.) If the recipient can't be found, it returns null.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMSER Server name. Must be the full name, starting with "S."

14. Mail Group Activities

- **CHK^XMA21**

Checks to see if a user is a member of a mail group. If the user is a member, **\$T** is set to "true"; otherwise, it's set to "false".



There is no other equivalent API.

Input Variables:

XMDUZ DUZ of the user in question.

Y IEN of the mail group in the MAIL GROUP file (#3.8).

Output Variables:

\$T

- **\$\$DM^XMBGRP(XMGROUP,.XMY,XMQUIET)**

Deletes local members from a mail group. Only local members can be deleted with this call. There is no API that deletes any other type of member. Returns 1, if no error; 0 otherwise.



There is no other equivalent API.

Input Parameters:

XMGROUP Mail group IEN or full name (without the **G.**).

.XMY Array of local users to delete from the mail group. Only DUZs are accepted (names are not).

XMQUIET Silent flag.

=0—interactive. Any errors will be displayed.

=1—silent (default). Any errors will be sent in a message to the Postmaster and the user (DUZ) making this call.

Parameters Killed Upon Exit:

XMY

- **\$\$MG^XMBGRP(XMGROUP,XMTYPE,XMORG,XMSELF,.XMY,.XMDESC,XMQUIET)**

Creates a new mail group or adds members to an existing mail group. Only local members can be added with this call. There is no API that adds any other type of member. Returns mail group IEN if no error; 0 otherwise.



There is no other equivalent API.

Input Parameters:

XMGROUP	=Mail group name (without the G.), if new mail group. =Mail group IEN or full name (without the G.), if existing mail group.
XMTYPE	Type of mail group. Used only for creating a mail group, otherwise it's ignored. =0—Public (default) =1—Private
XMORG	DUZ of group organizer. Used only for creating a mail group, otherwise it's ignored.
XMSELF	Allow self-enrollment? (0=no; 1=yes) Used only for creating a mail group, otherwise it's ignored.
.XMY	Array of local users to add to the mail group. Only DUZs are accepted (names are not).
.XMDESC	Array of text to put in the description field of the mail group. Used only for creating a mail group, otherwise it's ignored.
XMQUIET	Silent flag. =0—Interactive. Any errors will be displayed. =1—Silent (default). Any errors will be sent in a message to the Postmaster and the user (DUZ) making this call.

Parameters Killed Upon Exit:

XMY

PART II: New MailMan APIs (Post-Patch 50)

("NOSEND")	User may or may not send messages (multiple logon): 0—may send messages 1—may <i>not</i> send messages
("ORDER")	User's preferred message display order: 1—chronological -1—reverse chronological
("PRIV")	Contains surrogate privileges: "R"—Read "W"—Write
("RDR ASK")	Ask user which message reader to use? 0—no 1—yes
("RDR DEF")	User's default message reader: C—Classic D—Detailed full screen S—Summary full screen
("SHOW INST")	Show user's institution? 0—no 1—yes
("SHOW TITL")	Show user's title? 0—no 1—yes
("SYSERR",i)	<text> Domain incorrectly set up.
("VERSION")	"VA MailMan "_<version number>
("WARNING",1)	"Priority Mail"
("WARNING",2)	"Message in Buffer"
("WARNING",3)	"No Introduction"
("WARNING",4)	"Multiple Signon"
("WARNING",5)	"Postmaster has "_I_" baskets." (if more than 900)

XMDISPI	<p>[^]" Piece 1 contains any of following:</p> <ul style="list-style-type: none"> ➤ "T"—Show titles ➤ "A"—Ask basket ➤ "I"—Show institution <p>[^]" Piece 2 contains either of following message action defaults:</p> <ul style="list-style-type: none"> ➤ "I"—Ignore ➤ "D"—Delete
XMDUN	Same as XMV("NAME").
XMNOSEND	=1 if XMV("NOSEND")=1, otherwise not defined.
XMPRIV	Surrogate READ (y/n) [^] WRITE (send, y/n) privilege. Example: "y [^] n" means that the user may read, but <i>not</i> send messages.

- **OTHER[^]XMVVITAE**

Change certain MailMan settings when the user becomes a surrogate.



Meant to be called whenever the user becomes a surrogate.

Input Variables:

DUZ	User DUZ.
XMDUZ	Surrogate DUZ.

Output Variables:

XMV	Array of values, all pertaining to the user signified by XMDUZ. Any XMV variables not listed here are unchanged.
("LAST USE")	Date/time user last entered MailMan: dd mmm yy hh:mm
("NAME")	Name of user.
("NETNAME")	User's name as it will appear on messages sent to remote sites.
("NEW MSGS")	Number of new messages.
("NOSEND")	User may (=0) or may not (=1) send messages (multiple logon).
("PRIV")	Contains surrogate privileges ("R" READ, "W" WRITE).

The following XMV variables are KILLED and then set only if conditions warrant:

("BANNER")	User's banner.
("ERROR",1)	"You do not have a DUZ."
("ERROR",2)	"There is no person with DUZ "_XMDUZ_"."
("ERROR",3)	"There is no Access Code for DUZ "_XMDUZ_"."
("ERROR",4)	"There is no Mail Box for DUZ "_XMDUZ_"."
("WARNING",1)	"Priority Mail".
("WARNING",2)	"Message in Buffer".
("WARNING",3)	"No Introduction".
("WARNING",4)	"Multiple Signon".
("WARNING",5)	"POSTMASTER has "_I_" baskets." (if more than 900).

XMDUN Same as XMV("NAME").

XMNOSEND =1 if XMV("NOSEND")=1, otherwise not defined.

XMPRIV Surrogate READ (y/n) "^" WRITE (send, y/n) privilege.

Example: "y^n" means that the user may read, but *not* send messages.

- **SELF^XMOVVITAE**

Restore certain MailMan settings when the user becomes himself or herself again, after having been a surrogate.



Meant to be called whenever the user returns from being a surrogate.

Input Variables:

DUZ User DUZ.

Output Variables:

XMDUZ User DUZ.

XMV Array of values, all pertaining to the user signified by DUZ. Any XMV variables not listed here are unchanged.

("NAME") Name of user.

("NETNAME") User's name as it will appear on messages sent to remote sites.

("NEW MSGS") Number of new messages.

("NOSEND") User may (=0) or may not (=1) send messages (multiple logon).

The following XMV variables are KILLED and then set only if conditions warrant:

("BANNER") User's banner.

("ERROR",1) "You do not have a DUZ."

("ERROR",2) "There is no person with DUZ "_XMDUZ_"."

("ERROR",3) "There is no Access Code for DUZ "_XMDUZ_"."

("ERROR",4) "There is no Mailbox for DUZ "_XMDUZ_"."

("WARNING",1) "Priority Mail".

("WARNING",2) "Message in Buffer".

("WARNING",3) "No Introduction".

("WARNING",4) "Multiple Signon".

("WARNING",5) "POSTMASTER has "_I_" baskets." (if more than 900).

XMDUN Same as XMV("NAME").

XMNOSEND =1 if XMV("NOSEND")=1, otherwise not defined.

Variables Killed:

XMV("PRIV"), XMPRIV

16. Message Actions—^XMXAPI

Errors

If any errors occur, the following variables will be defined:

XMERR The number of errors.

^TMP("XMERR", \$J, <error number>, "TEXT", <line number>) = <error text>

Parameter Definitions

XMINSTR (optional) Array of special instructions:

("ADDR FLAGS") Special addressing instructions, any or all of the following:

I—Do not Initialize (KILL) the ^TMP addressee global, because it already contains addressees for this message, as a result of a previous call to an API.

R—Do not Restrict message addressing:

- Ignore "domain closed."
- Ignore "keys required for domain."
- Ignore "may not forward to domain."
- Ignore "may not forward priority mail to groups."
- Ignore "message length restrictions to remote addressees."

X—Do not create the ^TMP addressee global, because addressees are only being checked for validity.

("FLAGS") Message is any or all of the following:

P—Priority.

I—Information Only (may *not* be replied to).

X—Closed message (may *not* be forwarded).

C—Confidential message (surrogate may *not* read).

S—Send to sender (make sender a recipient).

R—Confirm receipt (return receipt requested).

("FROM") String saying who the message is from (default is user). This string is placed in Field #1, "From," in the MESSAGE file (#3.9). It must *not* be any real person, except for the Postmaster. The DUZ is *not* captured in

	the SENDER field (#1.1) in the MESSAGE file, thus, making this option well suited for messages from VISTA packages.
("FWD BY")	String saying who forwarded the message (default is user). This string is placed in Field #8, "Forwarded by," in the recipient multiple of the MESSAGE file (#3.9). It must <i>not</i> be any real person, except for the Postmaster. The DUZ is <i>not</i> captured in Field #8.01, "Forwarded by (xmduz)" in the recipient multiple of the MESSAGE file, thus, making this option well-suited for messages forwarded by VISTA packages.
("HDR")	Print the messages with a header? 0—no 1—yes (default).
("LATER")	Date/time (any format understood by VA FileMan) on which to send this message. Default is now.
("NET REPLY")	Should reply be sent over the network? 0—no (default) 1—yes Currently, only valid if sender of original message is remote.
("NET SUBJ")	Subject of reply to be sent over the network. Default is: "Re: <subject of original message>" Ignored unless XMINSTR("NET REPLY")=1.
("RCPT BSKT")	Basket to deliver to for all recipients. Default is the "IN" basket. Recipients must have specified in their personal preferences that such targeted basket delivery is allowed. Otherwise, this option is ignored.
("RECIPS")	Print recipients along with the message? 0—No (default). 1—Print summary recipients. 2—Print detailed recipients.
("RESPS")	Print which responses? *—Original message and all responses (default). 0—Original message only. range list (e.g., "0-3,5,7-99")—Ignored if more than one message is printed. This parameter is not checked. It <i>must</i> be correct. Range list may also be open-ended (e.g., "1,2,5-")

means print responses 1, 2, and responses 5 to the end).

("SCR KEY")	Scramble key (implies that message should be scrambled). It <i>must</i> be from 3 to 20 characters in length.
("SCR HINT")	Hint for scramble key (mandatory if message is to be scrambled). It <i>must</i> be from 1 to 40 characters in length.
("SELF BSKT")	Basket to deliver to, if sender is recipient. Default is the "IN" basket.
("SHARE BSKT")	Basket to deliver to if SHARED,MAIL is recipient. Default is the "IN" basket.
("SHARE DATE")	Date/time (any format understood by VA FileMan) to delete this message from SHARED,MAIL if SHARED,MAIL is the recipient.
("STRIP")	String containing characters to strip from the message text (XMBODY). It <i>must</i> be from 1 to 20 characters in length.
("TO PROMPT")	During interactive message addressing, contains the suggested initial addressee. Default is the user identified by XMDUZ.
("TYPE")	Message type is one of the following special types: D—Document S—Spooled Document X—DIFROM O—ODIF B—BLOB K—KIDS
("VAPOR")	Date/time (any format understood by VA FileMan) on which to delete (vaporize) this message from recipient baskets. Recipients may override this date.
("WHEN")	Date/time (any format understood by VA FileMan) on which to print messages. Default is now.

[.]XMTO Addressee or addressee array. If it is an array, it *must* be passed by reference.

- User's DUZ, or enough of user's name for a positive ID. For example:
1301, "lastname,first", ARRAY(1301)="", or
ARRAY("lastname,first")=""
- **G.group name** (enough for positive ID)
- **S.server name** (enough for positive ID)

- **D.device name** (enough for positive ID)

Prefix the above (except devices and servers) by:

I:—For "Information Only" recipient (may not reply). For example:

"I:1301" *or*

"I:lastname,first"

C:—For "copy" recipient (not expected to reply). For example:

"C:1301" *or*

"C:lastname,first"

L@datetime:—For when (in future) to send to this recipient (datetime can be anything accepted by VA FileMan). For example:

"L@25 DEC@0500:1301" *or*

"L@1 JAN:lastname,first" *or*

"L@2981225.05:1301"

(May combine **IL@datetime:** or **CL@datetime:**)

To delete recipients, prefix the recipients' name with a minus sign ("-"). For example:

-1301 *or*

"-lastname,first"

To address any recipient (including users, groups, devices, and servers) at a remote site, just add the @site name. For example:

recipient@site name

XMK and XMKZ for APIs which act on one message:

XMK (optional, depending on XMKZ) Basket (IEN or name) containing message.

XMKZ Identifies the message. Either:

- Message number (XMZ) in the MESSAGE global (XMK must *not* be specified).
- Message sequence number in the basket (XMK *must* be specified).

XMK and [.]XMKZA for APIs which act on groups of messages:

- | | |
|----------|--|
| XMK | (optional, depending on XMKZA) Basket (IEN or name) containing messages. |
| [.]XMKZA | Identifies messages, using a list or list array, which may end in a comma. Either: <ul style="list-style-type: none"> ➤ Message numbers (XMZ) in the MESSAGE global (XMK must <i>not</i> be specified <i>and</i> ranges are <i>not</i> allowed): <ul style="list-style-type: none"> - List: "1234567" or "1234567,9763213" - List array: ARRAY(1234567)=""
ARRAY(9763213)="" ➤ Message numbers in the basket (XMK <i>must</i> be specified and ranges <i>are</i> allowed): <ul style="list-style-type: none"> - List: "1" or "1,3,5-7" - List array: ARRAY("1,3")=""
ARRAY("5-7")="" |

- **ANSRMSG^XMXAPI(XMDUZ,XMK,XMKZ,XMSUBJ,XMBODY,[.]XMTO,.XMINSTR,.XMZR)**

Answer a message. (Send a new message, with a copy of the original message, to the sender of the original message). Sandwiches the user's answer (XMBODY) between the copy of the original message and a copy of the user's Network Signature. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate with "WRITE" privilege may use this API. SHARED,MAIL may not answer a message.

Input Parameters:

- | | |
|-----------|---|
| XMDUZ | The user (DUZ or name) who is answering a message. |
| XMK, XMKZ | Message being answered. For a description of these parameters, please refer to the "Parameter Definitions" list above. |
| XMSUBJ | (optional) Subject of answer. It <i>must</i> be from 3 to 65 characters in length. If null, it defaults to:

"Re: <subject of original message>" |
| XMBODY | Closed root of array containing answer text. The root may <i>not</i> be called "XMBODY". Also, it <i>must</i> be VA FileMan WORD-PROCESSING compatible. |
| [.]XMTO | (optional) Additional recipients of answer. (Answer is automatically addressed to sender of original message.) Passed by reference or by value. |

.XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above:
"ADDR FLAGS", "FLAGS", "FROM", "LATER", "RCPT BSKT", "SCR HINT", "SCR KEY", "SELF BSKT", "SHARE BSKT", "SHARE DATE", "STRIP", "TYPE", "VAPOR"

Output Parameters:

.XMZR Message number (XMZ) of the answer in the MESSAGE global.

- **DELMMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,.,XMMSG)**

Delete messages from a basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be deleted.

XMK,[.]XMKZA Messages to delete. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

Output Parameters:

.XMMSG If deletion is completed successfully, contains the message:
"<number of messages> deleted"

- **FLTRMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,.,XMMSG)**

Filter messages in a basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be filtered.

XMK,[.]XMKZA Messages to filter. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

Output Parameters:

.XMMSG If filter is completed successfully, contains the message:
 "<number of messages> filtered"

- **FWDMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,[.]XMTO,.,XMINSTR,.,XMMSG)**

Forward messages from a basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be forwarded.

XMK,[.]XMKZA Messages to forward. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

[.]XMTO To whom. For a description of this parameter, please refer to the "Parameter Definitions" list above. Passed by reference or by value.

.XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above:
 "ADDR FLAGS", "FWD BY", "LATER", "SELF BSKT", "SHARE BSKT",
 "SHARE DATE"

Output Parameters:

.XMMSG If forward is completed successfully, contains the message:
 "<number of messages> forwarded"

- **LATERMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,.,XMINSTR,.,XMMSG)**

"Later" messages in a basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be latered.

XMK,[.]XMKZA Messages to later. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

.XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above:
"LATER"

Output Parameters:

.XMMSG If later is completed successfully, contains the message:
"<number of messages> latered"

- **MOVEMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,XMKTO,.XMMSG)**

Move messages to a/another basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be moved.

XMK,[.]XMKZA Messages to move. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

XMKTO Basket (IEN or name) to which to move the messages. The basket must already exist.

Output Parameters:

.XMMSG If move is completed successfully, contains the message:
"<number of messages> saved"

- **PRTMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,XMPRTTO,.XMINSTR,.XMMSG,
.XMTASK,[XMSUBJ],[.XMTO])**

Print messages. (Actually, creates a task to print them.) It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) whose messages are to be printed.
XMK,[.]XMKZA	Messages to print. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.
XMPRTTO	Name of printer on which to print messages. This parameter is <i>not</i> checked, and <i>must</i> be correct.
.XMINSTR	(optional) Appropriate special instructions For a description of this parameter, please refer to the "Parameter Definitions" list above: "HDR", "RECIPS", "RESPS", "WHEN"

The following input parameters are only applicable if XMPRTTO is a P-MESSAGE device, and even then, they are optional:

XMSUBJ	(optional) Subject of the P-MESSAGE message. (Default is "Queued Mail Report from <user name>.", where <user name> is XMV("NAME").)
.XMTO	(optional) Additional recipients of P-MESSAGE message. (Message is automatically addressed to XMDUZ.)

Output Parameters:

.XMMSG	If print is tasked successfully, contains the message, "<number of messages> printed."
.XMTASK	If print is tasked successfully, contains the value of ZTSK.

- **PUTSERV^XMXAPI(XMKN,XMZ)**

Put a message in a server basket. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.

Input Parameters:

XMKN Full server name, including "S."
XMZ Message number in the MESSAGE global.

Output Parameters:

None.

- **REPLYMSG^XMXAPI(XMDUZ,XMK,XMKZ,XMBODY,.,XMINSTR,.,XMZR)**

Reply to message (Attach reply to original message). It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) who is replying to a message.
XMK, XMKZ Message being replied to. For a description of these parameters, please refer to the "Parameter Definitions" list above.
XMBODY Closed root of array containing reply text. The root may *not* be called "XMBODY". Also, it *must* be VA FileMan WORD-PROCESSING compatible.
.XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above:
 "ADDR FLAGS", "FROM", "NET REPLY", "NET SUBJ", "SCR HINT", "SCR KEY", "STRIP"

Output Parameters:

.XMZR Message number (XMZ) of the reply in the MESSAGE global.

- **SENDERBULL^XMXAPI(XMDUZ,XMBNAME,.XMPARM,XMBODY,[.]XMTO,
.XMINSTR,.XMZ,.XMATTACH)**

Send a bulletin (returns XMZ). It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) who is sending the bulletin.
XMBNAME	Full name of the bulletin.
.XMPARM	(optional, unless parameters exist for the bulletin): XMPARM(<number>)=<value for parameter number>
XMBODY	(optional) Closed root of array containing additional bulletin text to append onto text predefined in bulletin. The root may <i>not</i> be called "XMBODY". Also, it <i>must</i> be VA FileMan WORD-PROCESSING compatible.
[.]XMTO	(optional) Additional recipients of the bulletin (in addition to those predefined in the bulletin). Passed by reference or by value.
.XMINSTR	(optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above: "ADDR FLAGS", "FLAGS", "FROM", "LATER", "RCPT BSKT", "STRIP", "TYPE", "VAPOR"
.XMATTACH	(optional) Array of files to attach to the bulletin. (Reserved for future use.)

Output Parameters:

.XMZ	Message number (XMZ) of the bulletin in the MESSAGE global.
------	---

- **SENDMSG^XMXAPI(XMDUZ,XMSUBJ,XMBODY,[.]XMTO,.XMINSTR,.XMZ,.XMATTACH)**

Send a message. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) who is sending the bulletin.
XMSUBJ	Subject of the message. It <i>must</i> be from 3 to 65 characters in length. If null, it defaults to "* No Subject * ". If the subject is "* No Subject * " and the message is sent to a remote site, the subject in the "SUBJECT:" header record will be null.
XMBODY	Closed root of array containing message text. The root may <i>not</i> be called "XMBODY". Also, it <i>must</i> be VA FileMan WORD-PROCESSING compatible.
[.]XMTO	Recipients of the message. For a description of this parameter, please refer to the "Parameter Definitions" list above. Passed by reference or by value.
.XMINSTR	(optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above: "ADDR FLAGS", "FLAGS", "FROM", "LATER", "RCPT BSKT", "SCR HINT", "SCR KEY", "SELF BSKT", "SHARE BSKT", "SHARE DATE", "STRIP", "TYPE", "VAPOR"
.XMATTACH	(optional) Array of files to attach to the message. (Reserved for future use.)

Output Parameters:

.XMZ	Message number (XMZ) of the message in the MESSAGE global. However, if \$D(XMINSTR("LATER")), then XMZ contains the task number of the task that will create the message at the specified later date.
------	---

- **TASKBULL^XMXAPI(XMDUZ,XMBNAME,.XMPARM,XMBODY,[.]XMTO,
.XMINSTR,.XMTASK,.XMATTACH)**

Send a bulletin (quicker than SENDBULL, but does *not* return XMZ). It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) who is sending the bulletin.
XMBNAME	Full name of the bulletin.
.XMPARM	(optional, unless parameters exist for the bulletin): XMPARM(<number>)=<value for parameter number>
XMBODY	(optional) Closed root of array containing additional bulletin text to append onto text predefined in bulletin. The root may <i>not</i> be called "XMBODY". Also, it <i>must</i> be VA FileMan WORD-PROCESSING compatible.
[.]XMTO	(optional) Additional recipients of bulletin (in addition to those predefined in bulletin). Passed by reference or by value.
.XMINSTR	(optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above: "ADDR FLAGS", "FLAGS", "FROM", "LATER", "RCPT BSKT", "STRIP", "TYPE", "VAPOR"
.XMATTACH	(optional) Array of files to attach to the bulletin. (Reserved for future use.)

Output Parameters:

.XMTASK	Task number (ZTSK) of the task that will create and send the bulletin.
---------	--

- **TERMMSG^XMXAPI(XMDUZ,XMK,[.]XMKZA,.XMMSG)**

Terminate messages, possibly from a basket. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose messages are to be terminated.

XMK,[.]XMKZA Messages to terminate. For a description of these parameters, please refer to the "Parameter Definitions" list above. XMKZA is passed by reference or by value.

Output Parameters:

.XMMSG If terminate is completed successfully, contains the message:
 "<number of messages> terminated"

- **ZAPSERV^XMXAPI(XMKN,XMZ)**

Delete a message from a server basket. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.

Input Parameters:

XMKN Full server name, including "S."

XMZ Message number in the MESSAGE global.

Output Parameters:

None.

Message Action Building Blocks—^XMXAPI

- **ADDRNSND^XMXAPI(XMDUZ,XMZ,[.]XMTO,.XMINSTR)**

Address and send a message (does *not* handle the message body). It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) who is sending the message.
XMZ	Message number in the MESSAGE global.
[.]XMTO	Recipients of the message. For a description of this parameter, please refer to the "Parameter Definitions" list above. Passed by reference or by value.
.XMINSTR	(optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above: "ADDR FLAGS", "FLAGS", "FROM", "LATER", "RCPT BSKT", "SCR HINT", "SCR KEY", "SELF BSKT", "SHARE BSKT", "SHARE DATE", "TYPE", "VAPOR"

Output Parameters:

None.

- **CRE8XMZ^XMXAPI(XMSUBJ,.XMZ)**

Create a new message stub in the MESSAGE file (#3.9). It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.

Input Parameters:

XMSUBJ	Subject of the message. It <i>must</i> be from 3 to 65 characters in length. If null, it defaults to " * No Subject * ". If the subject is " * No Subject * " and the message is sent to a remote site, the subject in the "SUBJECT:" header record will be null.
--------	---

Output Parameters:

.XMZ	Message number in the MESSAGE global.
------	---------------------------------------

- **TOWHOM^XMXAPI(XMDUZ,XMZ,XMTYPE,XMTO,.XMINSTR,.XMFULL)**

Check one message addressee. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.

Input Parameters:

XMDUZ	The user (DUZ or name) who is addressing the message.
XMZ	Message number in the MESSAGE global. Not necessary, if XMTYPE="S" and XMINSTR("ADDR FLAGS") contains "R".
XMTYPE	Determines what prompts are used with the user: S—User is sending a message. F—User is forwarding a message.
XMTO	One recipient of the message. For a description of this parameter, please refer to the "Parameter Definitions" list above. Must be passed by value.
.XMINSTR	(optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list above: "ADDR FLAGS"

Output Parameters:

.XMFULL	Full name of the recipient.
---------	-----------------------------

- **VSUBJ^XMXAPI(.XMSUBJ)**

Validate a subject. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



*If the subject is "*** No Subject ***" and the message is sent to a remote site, the subject in the "SUBJECT:" header record will be null.*

Input Parameters:

.XMSUBJ	Subject of the message. It <i>must</i> be from 3 to 65 characters in length.
---------	--

Output Parameters:

.XMSUBJ	Subject of message. Leading and trailing blanks are removed, as are control characters. Any sequence of three or more consecutive blanks is reduced to two. If the subject is null, it defaults to " * No Subject * ".
---------	---

17. Mailbox Actions—^XMXAPIB

Errors

If any errors occur, the following variables will be defined:

XMERR The number of errors.

^TMP("XMERR", \$J, <error number>, "TEXT", <line number>) = <error text>

- **CRE8MBOX^XMXAPIB(XMDUZ, XMDATE)**

Create a mailbox for a user. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.

Input Parameters:

XMDUZ The user (DUZ or name) for whom the mailbox is to be created.

XMDATE (optional) Users who are being reinstated after not having worked here for a while may be restricted from seeing messages earlier than a certain date. If the user is a first-time user, then this parameter has no effect and shouldn't be used.

=0 or null—The user may access any message on the system that was ever addressed to the user.

=date—The user may not access any message addressed to the user on the system earlier than this date unless it is already in the user's mailbox or if someone forwards it to the user.

Output Parameters:

None.

- **FLTRMBOX^XMXAPIB(XMDUZ, XMMSG)**

Filter all messages in a user's mailbox. Runs all messages in all baskets in the user's mailbox through any filters that may exist for the mailbox. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose mailbox (all baskets) is to be filtered.

Output Parameters:

.XMMSG If filtering is completed successfully, contains the message, "Mailbox filtered".

- **QMBOX^XMXAPIB(XMDUZ,.XMMSG)**

Query a mailbox for new messages. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose mailbox is to be queried.

Output Parameters:

.XMMSG If no new messages, string is 0. If there are new messages, string contains the following pieces of information, separated by an up-arrow("^"):

Piece 1: Number of new messages in the mailbox.

Piece 2: Does the user have new priority mail?

0—no

1—yes

Piece 3: Number of new messages in the "IN" basket.

Piece 4: Date/time (in VA FileMan format) that the last message was received.

Piece 5: Have there been any new messages since the last time this routine was called:

0—no

1—yes

- **TERMMBOX^XMXAPIB(XMDUZ)**

Remove all traces of a user from MailMan globals. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.



Only a Postmaster surrogate or users who hold the XMMGR security key may use this API.

Input Parameters:

XMDUZ The user (DUZ or name) whose mailbox is to be terminated.

Output Parameters:

None.

18. Basket Actions—^XMXAPIB

Errors

If any errors occur, the following variables will be defined:

XMERR The number of errors.

^TMP("XMERR", \$J, <error number>, "TEXT", <line number>)=<error text>

- **CRE8BSKT^XMXAPIB(XMDUZ,XMKN,.XMK)**

Create a basket. If the basket already exists, an error is returned. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API. If the user is SHARED,MAIL, then the surrogate must be a Postmaster surrogate or hold the XMMGR security key.

Input Parameters:

XMDUZ The user (DUZ or name) for whom a basket is to be created.

XMKN The name of the basket. Basket name is FREE TEXT. It can be from 2 to 30 characters in length.

Output Parameters:

.XMK Basket number. The IEN of the basket created.

- **DELBSKT^XMXAPIB(XMDUZ,XMK,XMFLAGS)**

Delete a user's basket. The special baskets ("IN" and "WASTE") *cannot* be deleted. Only empty baskets can be deleted, unless XMFLAGS contains "D." It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API. If the user is SHARED,MAIL, then the surrogate must be a Postmaster surrogate or hold the XMMGR security key.

Input Parameters:

XMDUZ The user (DUZ or name) for whom a basket is to be deleted.

XMK Basket (IEN or name) to be deleted.

XMFLAGS (optional) Used to control processing:

 D—Delete the basket even if there are messages in it.

Output Parameters:

None.

- **FLTRBSKT^XMXAPIB(XMDUZ,XMK, XMMSG)**

Filter messages in a basket. Runs all messages in a basket through any filters which may exist for the mailbox. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API. If the user is SHARED,MAIL, then the surrogate must be a Postmaster surrogate or hold the XMMGR security key.

Input Parameters:

XMDUZ The user (DUZ or name) whose basket is to be filtered.

XMK Basket (IEN or name) to be filtered.

Output Parameters:

.XMMSG If filtering is completed successfully, contains the message, "Basket filtered".

- **LISTBSKT^XMXAPIB(XMDUZ,XMFLAGS,XMAMT,.XMSTART,XMPART,XMTROOT)**

Get a list of baskets in a mailbox. Gets a list (similar in format to that produced by LIST^DIC) of a user's baskets, optionally restricting the list to only those baskets with new mail, and/or those baskets whose name starts with a certain string. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Regardless of the alphabetic order you request, lowercase names sort separately from uppercase names. Therefore, an all uppercase cross-reference (under "BSKT") is provided, if you do not limit the number of entries returned.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) for whom a basket list is to be compiled.
XMFLAGS	(optional) Used to control processing: B—Backwards alpha order (default is alpha order). N—Baskets with new messages only.
XMAMT	(optional) How many? Number—Get this many. *—Get all (default) and provide uppercase basket name cross-reference.
.XMSTART	(optional) Used to start the Lister going. The Lister will keep it updated from call to call.
XMPART	(optional) List only those baskets whose name starts with this string.
XMTROOT	Target root (closed) to receive the message list. (default is ^TMP("XMLIST", \$J)).

Output Parameters:

None.

Example:

In the following example, notice that:

- The node "XMLIST" has been added under the target root that was specified.
- The header node of the list is identical to the header node produced by VA FileMan's LIST^DIC.
- The list is in alphabetical order. Unlike VA FileMan's LIST^DIC, the counter starts at 1, whether you've requested forward or reverse order, and when you traverse the list, starting at 1, you are traversing in the order you requested.
- The data for each basket is the same as that produced by QBSKT^XMXAPIB, namely:
 - Piece 1: Basket IEN.
 - Piece 2: Basket name.
 - Piece 3: Number of messages in the basket.
 - Piece 4: Number of new messages in the basket.

```
>K XMSTART
>D LISTBSKT^XMXAPIB(3,,6,.XMSTART,, "GARY")
>ZW GARY
GARY("XMLIST",0)=6^6^1
GARY("XMLIST",1)=10^98 BASKET^1^
GARY("XMLIST",2)=8^DELIVERY^1^
GARY("XMLIST",3)=2^FRED^4^0
GARY("XMLIST",4)=9^FREDOLINA^0^
GARY("XMLIST",5)=1^IN^335^1
GARY("XMLIST",6)=4^JOHN^10^0

>ZW XMSTART
XMSTART=JOHN
XMSTART("IEN")=4

>D LISTBSKT^XMXAPIB(3,,5,.XMSTART,, "GARY")
>ZW GARY
GARY("XMLIST",0)=5^5^0
GARY("XMLIST",1)=3^JOSE^9^0
GARY("XMLIST",2)=6^KITTY^13^0
GARY("XMLIST",3)=5^MARIA^4^0
GARY("XMLIST",4)=7^SCRAMBLE^3^0
GARY("XMLIST",5)=.5^WASTE^44^

>ZW XMSTART
XMSTART=
XMSTART("IEN")=
```

- **LISTMSG^XMXAPIB(XMDUZ,XMK,XMFLDS,XMFLAGS,XMAMT,.XMSTART,.XMCRT,XMTROOT)**

Get a list of messages in a mailbox. Gets a list (similar in format to that produced by LIST^DIC) of messages in one basket or all baskets, optionally based on certain criteria. The IENs of the messages in the MESSAGE file (#3.9) are returned. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) for whom a message list is to be compiled.
XMK	Basket to look in IEN or name—Look in this basket <i>only</i> . *—Look in all baskets.
XMFLDS	(optional) A string containing a list of fields to retrieve, separated by ";" (default is none).

For example, XMFLDS="SUBJ;DATE" means retrieve subject and date.

"BSKT"	Basket (default: <basket IEN>^<basket name>) Optionally followed by ":" and "I" for basket IEN only (no 2 nd piece). "X" adds basket name cross-reference.
"DATE"	Date sent (default: <internal date>^<dd mmm yy hh:mm>). Optionally followed by ":" and "I" for internal only (no 2 nd piece). "F" for VA FileMan date as the 2 nd piece. "X" adds VA FileMan date cross-reference.
"FROM"	Message from (default: <internal from>^<external from>) Optionally followed by ":" and "I" for internal only (no 2 nd piece) "X" adds external "From" cross-reference.
"LINE"	Number of lines in the message.
"NEW"	Is the message new? 0—no. 1—yes. 2—yes, and priority, too.

"PRI"	Is the message priority? 0—no. 1—yes.
"READ"	How much of the message has the user read? null—has not read the message at all. 0—has read the message, but no responses. number—has read through this response.
"RESP"	How many responses does the message have? 0—none. number—this many.
"SEQN"	Sequence number in basket.
"SUBJ"	Message subject (always external). Optionally followed by ":" and "X" adds message subject cross-reference.
XMFLAGS	(optional) Used to control processing B—Backwards order (default is traverse forward). C—Use basket C-cross-reference (default is message IEN). N—New messages only (C flag ignored). P—New Priority messages only (C, N flags ignored).
XMAMT	(optional) How many? Number—Get this many. *—Get all (default).
.XMSTART	(optional) Used to start the Lister going. The Lister will keep it updated from call to call: ("XMK") Start with this basket IEN (valid only if XMK="*"). Continues from there, with each successive call, to the end. (Default is to start with basket ".5"—the "WASTE" basket.) ("XMZ") Start <i>after</i> this message IEN (valid only if no C flag). Continues from there, with each successive call, to the end. (Default is to start at the beginning (or end) of the basket.) ("XMKZ") Start <i>after</i> this message sequence number (valid only if C flag). Continues from there, with each successive call, to the end. (Default is to start at the beginning [or end] of the basket.)

.XMCRIT	(optional) Criteria which are "and"ed together to select only those messages which meet the criteria. (Default is to get a list of all messages in the basket or mailbox.)
("FROM")	<p>Message is from this person.</p> <p>If the person is a local user, it <i>must</i> be the user's DUZ.</p> <p>If the person is a remote user, it <i>must</i> be <partial name>_"@"_<partial domain>.</p> <p>The search is <i>not</i> case sensitive.</p> <p>For example:</p> <p>"fred@"—Finds any person whose name contains "fred".</p> <p>"@va"—Finds any person whose domain contains "va".</p> <p>"fred@va"—Finds any person whose name contains "fred" and domain contains "va".</p>
("FDATE")	Message was sent on or after this date. Date <i>must</i> be in VA FileMan format.
("RFROM")	Message has a response from this person (same rules as "FROM").
("SUBJ")	Subject contains this string.
("SUBJ","C")	<p>Is search case sensitive?</p> <p>0—Search is not case sensitive (default).</p> <p>1—Search is case sensitive.</p>
("TDATE")	Message was sent on or before this date. Date <i>must</i> be in VA FileMan format.
("TEXT")	Message contains this string.
("TEXT","L")	<p>Where should we look for the text?</p> <p>1—Look in message only (default).</p> <p>2—Look in both message and responses.</p> <p>3—Look in responses only.</p>
("TEXT","C")	<p>Is search case sensitive?</p> <p>0—Search is not case sensitive (default).</p> <p>1—Search is case sensitive.</p>
("TO")	Message is to this person (same rules as "FROM").
XMTROOT	Target root (closed) to receive the message list. (Default is ^TMP("XMLIST",\$J).)

Output Parameters:

None.

Example:

```
>D LISTMSG^XMXAPIB(3,2,"FROM:X;SUBJ;DATE:IX","B")
```

In the following example, notice that:

- The list is in reverse XMZ order (because of the "B" flag). Unlike VA FileMan's LIST^DIC, the counter starts at one, whether you've requested forward or reverse order, and when you traverse the list, starting at one, you are traversing in the order you requested.
- The header node of the list is identical to the header node produced by VA FileMan's LIST^DIC.
- The cross-reference for the date field is adjusted to local time (PST, in this case) if the message is from a remote site.
- The external form of the from field removes the "<>" if the message is from a remote site.

```
^TMP("XMLIST",564222283,0) = 4^^^0
^TMP("XMLIST",564222283,1) = 978181
^TMP("XMLIST",564222283,1,"DATE") = 08 Apr 97 15:08 PST
^TMP("XMLIST",564222283,1,"FROM") = <FUCHSIA.GARY_M+@ISC-SF.VA.GOV>^
FUCHSIA.GARY_M+@ISC-SF.VA.GOV
^TMP("XMLIST",564222283,1,"SUBJ") = FRED
^TMP("XMLIST",564222283,2) = 977961
^TMP("XMLIST",564222283,2,"DATE") = 02 Dec 96 15:37 EST
^TMP("XMLIST",564222283,2,"FROM") = <MINTGREEN.MILT@FORUM.VA.GOV>^
MINTGREEN.MILT@FORUM.VA.GOV
^TMP("XMLIST",564222283,2,"SUBJ") = XMMILTKI UTILITY
^TMP("XMLIST",564222283,3) = 977827
^TMP("XMLIST",564222283,3,"DATE") = 2960809.10363
^TMP("XMLIST",564222283,3,"FROM") = 3^FUCHSIA,GARY
^TMP("XMLIST",564222283,3,"SUBJ") = TEST NEW PROTOCOL
^TMP("XMLIST",564222283,4) = 977647
^TMP("XMLIST",564222283,4,"DATE") = 2960725.142942
^TMP("XMLIST",564222283,4,"FROM") = .5^POSTMASTER
^TMP("XMLIST",564222283,4,"SUBJ") = G.FRED@ISC-SF.VA.GOV NOT FOUND
^TMP("XMLIST",564222283,"DATE",2960725.142942,4) =
^TMP("XMLIST",564222283,"DATE",2960809.10363,3) =
^TMP("XMLIST",564222283,"DATE",2961202.1337,2) =
^TMP("XMLIST",564222283,"DATE",2970408.1508,1) =
^TMP("XMLIST",564222283,"FROM","FUCHSIA,GARY",3) =
^TMP("XMLIST",564222283,"FROM","FUCHSIA.GARY_M+@ISC-SF.VA.GOV",1) =
^TMP("XMLIST",564222283,"FROM","POSTMASTER",4) =
^TMP("XMLIST",564222283,"FROM","MINTGREEN.MILT@FORUM.VA.GOV",2) =
```

- **NAMEBSKT^XMXAPIB(XMDUZ,XMK,XMKN)**

Change the name of a basket. The "IN" and "WASTE" baskets may *not* be renamed. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate with "WRITE" privileges may use this API. If the user is SHARED,MAIL, then the surrogate must be a Postmaster surrogate or hold the XMMGR security key.

Input Parameters:

XMDUZ	The user (DUZ or name) whose basket is to be renamed.
XMK	Basket (IEN or name) to be renamed.
XMKN	The new name of the basket. Basket name is FREE TEXT. It can be from 2 to 30 characters in length.

Output Parameters:

None.

- **QBSKT^XMXAPIB(XMDUZ,XMK,.XMMSG)**

Get information on a basket. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.



Only the user or a surrogate may use this API.

Input Parameters:

XMDUZ	The user (DUZ or name) whose basket is to be queried.
XMK	Basket (IEN or name) to be queried.

Output Parameters:

.XMMSG	String containing the following pieces of information, separated by an up-arrow ("^"):
	Piece 1: Basket IEN.
	Piece 2: Basket name.
	Piece 3: Number of messages in the basket.
	Piece 4: Number of new messages in the basket.

- **RSEQBSKT^XMXAPIB(XMDUZ,XMK,.XMMSG)**

Resequence messages in a basket. Before the resequencing is done, a basket integrity check is performed, and any errors detected are corrected. It sets XMERR and ^TMP("XMERR",\$J), if an error occurs.



Only the user or a surrogate may use this API. If the user is SHARED,MAIL, then the surrogate must be a Postmaster surrogate or hold the XMMGR security key.

Input Parameters:

XMDUZ The user (DUZ or name) whose basket is to be resequenced.

XMK Basket (IEN or name) to be resequenced.

Output Parameters:

.XMMSG If resequencing is completed successfully, contains the message:
 "Resequenced from 1 to <number of messages in basket>"

19. Mail Group Functions and Actions—^XMXAPIG

Errors

If any errors occur, the following variables will be defined:

XMERR The number of errors.

^TMP("XMERR", \$J, <error number>, "TEXT", <line number>)=<error text>

- **\$\$GOTLOCAL^XMXAPIG(XMGROUP,XMDAYS)**

This API is used to find out whether or not a mail group has any *active* local members. Messages can only be delivered to *active* local members of a mail group. Active members are described as having an Access Code and a mailbox. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.

It returns the following:

0—No *active* members in a local mail group.

1—Has *active* members in a local mail group.

Usage:

```
I '$$GOTLOCAL^XMXAPIG(XMGROUP,XMDAYS) D error
```

Input Parameters:

XMGROUP Mail group IEN or name (exact, case-sensitive).

XMDAYS (optional) Active members of the mail group must have used MailMan within the past number of days specified by XMDAYS. If XMDAYS is 0, null, or not supplied, it is ignored.

Output Parameters:

None.

Example:

```
I '$$GOTLOCAL^XMXAPIG("GROUP") D error
```

If the mail group named "GROUP" has no active local members, do an error routine to notify someone. Otherwise, go ahead and send the message.

Optionally, you may specify an additional constraint, that at least one member must have used MailMan in the last few days:

```
I '$$GOTLOCAL^XMXAPIG("GROUP",9) D error
```

If the mail group named "GROUP" doesn't have at least one active local member who has used MailMan in the last 9 days, do an error routine to notify someone. Otherwise, go ahead and send the message.

20. Interactive User Actions—**^XM** & **^XMXAPIU**

^XM

These APIs may be used to create menu options.

The primary option should be set up as follows:

Entry action: **S XMMENU(0)=<name of the menu option> D EN^XM**

Routine: **xxx^XMXAPIU**

Exit action: **K XMMENU D CHECKOUT^XM**

Any subordinate option should be set up as follows:

Entry action: **D CHECKIN^XM**

Routine: **xxx^XMXAPIU**

Exit action: **D CHECKOUT^XM**

- **EN^XM**

Meant to be the entry action of the primary MailMan option. It sets up the user's MailMan environment and calls **HEADER^XM** to greet the user.

- **CHECKIN^XM**

Meant to be the entry action of any subordinate MailMan option.

- **CHECKOUT^XM**

Meant to be the exit action of every MailMan option.

- **HEADER^XM**

Meant to be part of the entry action of the primary MailMan option, whether called by itself, or as part of another call, such as **EN^XM**. It displays a greeting to the user.

^XMXAPIU

The following are meant to be in an option's ROUTINE field. They expect that DUZ exists, and if the user is acting as a surrogate, that XMDUZ exists too. Otherwise, XMDUZ will be set to DUZ. If the XMV variables do not exist, INIT^XMVITAE will be called.

- **READ^XMXAPIU**

Read/Manage messages in your mailbox.



Only the user or a surrogate may use this API.

Input Variables:

XMDUZ (optional) The user whose mailbox is to be read. Default is DUZ.

XMV (optional) The user's variables.

Output Variables:

None.

- **READNEW^XMXAPIU**

Read new messages in your mailbox.



Only the user or a surrogate may use this API.

Input Variables:

XMDUZ (optional) The user whose new messages are to be read. Default is DUZ.

XMV (optional) The user's variables.

Output Variables:

None.

- **SEND^XM APIU**

Send a message.



Only the user or a surrogate with "WRITE" privileges may use this API.

Input Variables:

XMDUZ (optional) The user who is to send a message. Default is DUZ.

XMV (optional) The user's variables.

Output Variables:

None.

The following is meant to be used in a routine:

- **TOWHOM^XM APIU(XMDUZ, XMZ, XM TYPE, .XM INSTR)**

Ask user for message addressees.

Input Parameters:

XMDUZ The user (DUZ or name) who is addressing the message.

XMZ Message number in the MESSAGE global. Not necessary, if XM TYPE="S" and XM INSTR("ADDR FLAGS") contains "R".

XM TYPE Determines what prompts are used with the user:
S—User is sending a message.
F—User is forwarding a message.

.XM INSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"ADDR FLAGS", "TO PROMPT"

Output Parameters:

.XM INSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"SELF BSKT", "SHARE BSKT", "SHARE DATE"

The following global variables are created:

- ^TMP("XMY0",\$J) Addressee as entered by the user.
- ^TMP("XMY",\$J) Resulting addressee(s) as interpreted by MailMan.

21. Message Editing—[^]XMxEDIT

These entry points edit different parts of a message. They do *not* perform any checks to see whether it is appropriate to do so. That is the responsibility of the calling routine.

Generally, these entry points expect that the input parameters are correct. They also expect that the calling application has assumed a level of responsibility and already taken care of the following:

- `INIT^XMVVITAE` Has been called to set up the user's XMV array, with vital user information, user preferences, and, if the user is a surrogate, determining level of authorization.
- Determined that the user is authorized to see the message. If the message is in the user's mailbox, then that's enough. Otherwise, `$$ACCESS^XMSEC` should be used to determine authorization.
- `OPTMSG^XMSEC2` Has been called and has given its permission to edit the message or to toggle Information Only.



`$$EDIT^XMSEC2` will also let you know whether the user may edit the message.

- `OPTEDIT^XMSEC2` Has been called and has given its permission to edit the particular thing we are editing here.
- `INMSG2^XMUTIL2` Has been called to set XMINSTR. These routines expect that XMINSTR has been correctly set. They will change XMINSTR according to the item being edited.

Errors

If any errors occur, the following variables may be defined:

`XMERR` The number of errors.

`^TMP("XMERR", $J, <error number>, "TEXT", <line number>)=<error text>`

- `CLOSED^XMxEDIT(XMZ, XMINSTR, XMMSG)`

Toggle the message's "Closed" flag. It sets `XMERR` and `^TMP("XMERR", $J)`, if an error occurs.

Input Parameters:

`XMZ` Message IEN in the MESSAGE file (#3.9).

`.XMINSTR` Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:

"FLAGS"

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"
- .XMMSG Appropriate message, suitable for display to the user.

- **CONFID^XMXEDIT(XMZ,.XMINSTR,.XMMSG)**

Toggle the message's "Confidential" flag. It sets XMERR and ^TMP("XMERR", \$J), if an error occurs.

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"
- .XMMSG Appropriate message, suitable for display to the user.

- **CONFIRM^XMXEDIT(XMZ,.XMINSTR,.XMMSG)**

Toggle the message's "Confirm Receipt Requested" flag. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"
- .XMMSG Appropriate message, suitable for display to the user.

- **DELIVER^XMXEDIT(XMZ,XMDBSKT,.XMINSTR,.XMMSG)**

Set/Delete the message delivery basket for all users. It does *not* set XMERR and ^TMP("XMERR",\$J).

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- XMDBSKT New Delivery basket name:
="@" (At-sign, Shift-2 key on most keyboards), if you want to delete it.

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
("RCPT BSKT") Set to XMDBSKT or KILLED if XMDBSKT="@".
- .XMMSG Appropriate message, suitable for display to the user.

- **INFO^XMXEDIT(XMZ,.XMINSTR,.XMMSG)**

Toggle the message's "Information Only" flag. It does *not* set XMERR and ^TMP("XMERR",\$J).

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"
- .XMMSG Appropriate message, suitable for display to the user.

- **PRIORITY^XMXEDIT(XMZ,.XMINSTR,.XMMSG)**

Toggle the message's "Priority" flag. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"

Output Parameters:

- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS"
- .XMMSG Appropriate message, suitable for display to the user.

- **SUBJ^XMXEDIT(XMZ,XMSUBJ,.XMIM)**

Change the message subject. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

- XMZ Message IEN in the MESSAGE file (#3.9).
- XMSUBJ Subject of the message. It *must* be from 3 to 65 characters in length. If null, it defaults to "*** No Subject ***". If the subject is "*** No Subject ***" and the message is sent to a remote site, the subject in the "SUBJECT:" header record will be null.

Output Parameters:

- .XMIM Message information:
("SUBJ") Message subject.

- **TEXT^XMXEDIT(XMZ,XMBODY)**

Replace the message text. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

XMZ Message IEN in the MESSAGE file (3.9).

XMBODY Closed root of array containing new message text. The root may *not* be called "XMBODY". Also, it *must* be VA FileMan WORD-PROCESSING compatible.

Output Parameters:

None.

- **VAPOR^XMXEDIT(XMZ,XMVAPOR,.XMINSTR,.XMMSG)**

Set/Delete the message vaporize date. It does *not* set XMERR and ^TMP("XMERR", \$J).



This routine does not set the message vaporize date in a user's basket. Use KVAPOR^XMXUTIL to do that.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMVAPOR New message vaporize date/time. The date *must* be in VA FileMan format. = "@" (At-sign, Shift-2 key on most keyboards), if you want to delete it.

Output Parameters:

.XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
("VAPOR") Set to XMVAPOR or KILLED if XMVAPOR="@".

.XMMSG Appropriate message, suitable for display to the user.

22. Message Security, Permission, and Restriction Functions and Routines—[^]XM_XSEC

Errors

If any errors occur, the following variables will be defined:

`XMERR` The number of errors.
`^TMP("XMERR", $J, <error number>, "TEXT", <line number>)=<error text>`

[^]XM_XSEC

- `$$ACCESS^XMXSEC(XMDUZ, XMZ, XMZREC)`

Returns a value indicating whether the user may access a message or not (0=no; 1=yes). It sets `XMERR` and `^TMP("XMERR", $J)`, if access/permission is denied.

Input Parameters:

`XMDUZ` User DUZ.
`XMZ` Message IEN in the MESSAGE file (#3.9).
`XMZREC` (optional) Zero node of the message: `^XMB(3.9, XMZ, 0)`.

- `$$ANSWER^XMXSEC(XMDUZ, XMZ, XMZREC)`

Returns a value indicating whether the user may answer a message or not (0=no; 1=yes). It sets `XMERR` and `^TMP("XMERR", $J)`, if access/permission is denied.

Input Parameters:

`XMDUZ` User DUZ.
`XMZ` Message IEN in the MESSAGE file (#3.9).
`XMZREC` (optional) Zero node of the message: `^XMB(3.9, XMZ, 0)`.

- **\$\$BCAST^XMXSEC(XMZ)**

Returns a value indicating whether a message was broadcast or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$CLOSED^XMXSEC(XMZREC)**

Returns a value indicating whether a message is "Closed" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$ZCLOSED^XMXSEC described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$CONFID^XMXSEC(XMZREC)**

Returns a value indicating whether a message is "Confidential" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$ZCONFID^XMXSEC described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$CONFIRM^XMXSEC(XMZREC)**

Returns a value indicating whether a message is "Confirm Receipt Requested" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$ZCONFIRM^XMXSEC described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$COPY^XMXSEC(XMDUZ,XMZ,XMZREC)**

Returns a value indicating whether the user may copy a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR",\$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.
 XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: ^XMB(3.9,XMZ,0).

- **\$\$DELETE^XMXSEC(XMDUZ,XMK,XMZ,XMZREC)**

Returns a value indicating whether the user may delete or terminate a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR",\$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.
 XMK Basket IEN.
 XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: ^XMB(3.9,XMZ,0).

- **\$\$FORWARD^XMXSEC(XMDUZ,XMZ,XMZREC)**

Returns a value indicating whether the user may forward a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR",\$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.
 XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: ^XMB(3.9,XMZ,0).

- **\$\$INFO^XMXSEC(XMZREC)**

Returns a value indicating whether a message is "Information Only" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR", \$J).



Compare to \$\$ZINFO^XMXSEC described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$LATER^XMXSEC(XMDUZ)**

Returns a value indicating whether the user may "later" a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

- **\$\$MOVE^XMXSEC(XMDUZ, XMZ, XMZREC)**

Returns a value indicating whether the user may save or filter a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$ORIGIN8R^XMXSEC(XMDUZ, XMZREC)**

Returns a value indicating whether the user (XMDUZ or DUZ) sent the message or not (sender or surrogate, 0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR", \$J).



Compare to \$\$ZORIGIN8^XMXSEC described below.

Input Parameters:

XMDUZ User DUZ.

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$POSTPRIV^XMXSEC()**

Returns a value indicating whether the user has Postmaster privileges or not, including whether or not the user may perform group message actions in SHARED,MAIL (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

None.

- **\$\$PRIORITY^XMXSEC(XMZREC)**

Returns a value indicating whether a message is "Priority" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR", \$J).



Compare to \$\$ZPRI^XMXSEC described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$READ^XMXSEC(XMDUZ, XMZ, XMZREC)**

Returns a value indicating whether the user may read a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$REPLY^XMXSEC(XMDUZ, XMZ, XMZREC)**

Returns a value indicating whether the user may reply to a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$RPRIV^XMXSEC()**

Returns a value indicating whether the surrogate has READ privileges or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

None.

- **\$\$RWPRIV^XMXSEC()**

Returns a value indicating whether the surrogate has READ or WRITE privileges or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

None.

- **\$\$SEND^XMXSEC(XMDUZ, XMINSTR)**

Returns a value indicating whether the user may send a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

.XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FROM"

- **\$\$SURRACC^XMXSEC(XMDUZ, XMAccess, XMZ, XMZREC)**

Returns a value indicating whether the surrogate may access a message or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.

XMAccess String telling type of access attempted. (Used in an error message, if access is denied.)

XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$\$SURRCONF^XMXSEC(XMDUZ, XMZ)**

Returns a value indicating whether a message is "Confidential" or not, and if it is, whether the surrogate may access it (0=no; 1=yes, it is confidential and the surrogate may not access it). It does *not* set XMERR and ^TMP("XMERR", \$J).



This function should only be used when the user is a surrogate.

Input Parameters:

XMDUZ User DUZ.
 XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$WPRIV^XMXSEC()**

Returns a value indicating whether the surrogate has WRITE privileges or not (0=no; 1=yes). It sets XMERR and ^TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

None.

- **\$\$ZCLOSED^XMXSEC(XMZ)**

Returns a value indicating whether a message is "Closed" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR", \$J).



Compare to \$\$CLOSED^XMXSEC described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZCONFID^XMXSEC(XMZ)**

Returns a value indicating whether a message is "Confidential" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$CONFID^XMXSEC described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZCONFIRM^XMXSEC(XMZ)**

Returns a value indicating whether a message is "Confirm Receipt Requested" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$CONFIRM^XMXSEC described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZINFO^XMXSEC(XMZ)**

Returns a value indicating whether a message is "Information Only" or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$INFO^XMXSEC described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZORIGIN8^XMXSEC(XMDUZ,XMZ)**

Returns a value indicating whether the user (XMDUZ or DUZ) sent the message or not (sender or surrogate, 0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$ORIGIN8R^XMXSEC described above.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZPOSTPRV^XMXSEC()**

Returns a value indicating whether the user has Postmaster privileges or not, including whether or not the user may perform group message actions in SHARED,MAIL (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).

Input Parameters:

None.

- **\$\$ZPRI^XMXSEC(XMZ)**

Returns a value indicating whether a message is "Priority" or not (0=no;1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).



Compare to \$\$PRIORITY^XMXSEC described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

^XMXSEC1

- **\$\$COPYAMT^XMXSEC1(XMZ,XMWHICH)**

This function may be used when copying a message. It checks the total number of lines and responses to be copied. Returns 1 if the amount is within site limitations; 0, if not. It sets XMERR and ^TMP("XMERR", \$J), if permission is denied.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMWHICH (optional) String of which responses are to be copied. Options include:
 0—Original message.
 Number list/range—These particular responses.
 Undefined/Null—Original message and all responses.

- **\$\$COPYLIMS^XMXSEC1()**

This function may be used when copying a message. Returns (three-piece ^-delimited string) the site's message copy limits. If the site has no specific limit, then MailMan defaults are used. It does *not* set XMERR and ^TMP("XMERR", \$J).

- Piece 1: Number of recipients to whom the copy may be sent (default=2999).
- Piece 2: Number of responses that may be copied (default=99).
- Piece 3: Number of lines of text that may be copied (default=3999).

Input Parameters:

None.

- **\$\$COPYRECP^XMXSEC1(XMZ)**

This function may be used when copying a message. It checks the total number of recipients on the message to see if it's "OK" to list them in the copy text and send the copy to them, too. It returns 1 if the amount is within site limitations; 0, if not. It sets XMERR and ^TMP("XMERR", \$J), if permission is denied.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$PAKMAN^XMXSEC1(XMZ,XMZREC)**

Returns a value indicating whether a message is a PackMan message or not (0=no; 1=yes). It does *not* set XMERR and ^TMP("XMERR",\$J).

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).

- **CHKLINES^XMXSEC1(XMDUZ,XMZ,.XMRESTR)**

Checks whether a message is too long to be sent to a remote site. If \$D(XMRESTR("NONET")), then it is. It does *not* set XMERR and ^TMP("XMERR",\$J).



This routine does not KILL XMRESTR.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

Output Parameters:

If the message is not too long or if the user holds the XMMGR security key, then XMRESTR("NONET") is *not* set. Otherwise, XMRESTR("NONET") equals the maximum number of lines a message may have when sending to a remote site.

- **CHKMSG^XMXSEC1(XMDUZ,XMK,XMKZ,XMZ,XMZREC)**

Checks whether or not the message is located where the calling routine says it is, and whether or not the user may access it. It sets XMERR and ^TMP("XMERR",\$J), if access/permission is denied.

Input Parameters:

XMDUZ DUZ of the user who is accessing the message.

XMK, XMKZ Message being accessed. For a description of these parameters, please refer to the "Parameter Definitions" list in Chapter 16 in this manual.

Output Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **GETRESTR^XMXSEC1(XMDUZ, XMZ, XMZREC, .XMINSTR, .XMRESTR)**

Returns assorted restrictions, if any, on sending or forwarding the message. It does *not* set XMERR and ^TMP("XMERR", \$J).



This routine does not KILL XMRESTR.

Input Parameters:

XMDUZ User DUZ.
 XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: ^XMB(3.9, XMZ, 0).
 .XMINSTR (optional) Appropriate special instructions. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
 "ADDR FLAGS"

Output Parameters:

.XMRESTR Restrictions on forwarding the message. Here are the nodes that may be set:

- If \$G(XMRESTR("NONET")), then the message is too long to be sent to a remote site, and it equals the maximum number of lines a message may have when sending to a remote site.
- If \$G(XMRESTR("FLAGS"))["C", then the message may *not* be forwarded to SHARED,MAIL (because it is confidential).
- If \$G(XMRESTR("FLAGS"))["X", then the message may *not* be forwarded to SHARED,MAIL (because it is closed).
- If \$D(XMRESTR("NOFPG")), then the message may *not* be forwarded to groups (because it is a priority message, the user didn't send it, the user doesn't possess the XM GROUP PRIORITY security key, and XMINSTR("ADDR FLAGS")["R").

- **OPTGRP^XMXSEC1(XMDUZ,XMK,.XMOPT)**

Determines what the user may do at the basket or message group level. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

XMDUZ DUZ of the user who is accessing the basket.
 XMK Basket IEN.

Output Parameters:

.XMOPT Commands which the user may use are a subset of the following:
 If the user may not use them, they either won't appear (only the Postmaster will see "X") or they will have subnode(s) under "?" with an explanation as to why they may *not* be used.
 For example, if the basket were the "IN" basket, then XMOPT("C", "?")="The name of the IN basket may not be changed."
 If there were **n** lines of text, then XMOPT("C", "?", 1), ... XMOPT("C", "?", n-1), and XMOPT("C", "?") would be set.

("C")	"Change the name of this basket"
("D")	"Delete messages"
("F")	"Forward messages"
("FI")	"Filter messages"
("H")	"Headerless Print messages"
("L")	"Later messages"
("N")	"New messages list"
("P")	"Print messages"
("Q")	"Query (search for) messages"
("R")	"Resequene messages"
("S")	"Save messages"
("T")	"Terminate messages"
("X")	"Xmit priority toggle"

[^]XM_XSEC₂

- **\$\$EDIT[^]XM_XSEC₂(XMDUZ,XMZ,XMZREC)**

Returns a value indicating whether the user may edit a message or not (0=no; 1=yes). It sets XMERR and [^]TMP("XMERR", \$J), if access/permission is denied.

Input Parameters:

XMDUZ User DUZ.
 XMZ Message IEN in the MESSAGE file (#3.9).
 XMZREC (optional) Zero node of the message: [^]XMB(3.9,XMZ,0).

- **OPTEDIT[^]XM_XSEC₂(.XMINSTR,.XMOPT)**

If OPTMSG[^]XM_XSEC₂ determines that the user may edit the message, then OPTEDIT determines what, exactly, the user may edit. It does *not* set XMERR and [^]TMP("XMERR", \$J).

Input Parameters:

.XMINSTR Set by INMSG2[^]XM_XUTIL2:
 "FLAGS", "TYPE", "VAPOR", "RCPT BSKT", "SCR HINT"

Output Parameters:

.XMOPT Commands which the user may use are a subset of the following:
 If the user may not use them, they will have subnode(s) under "?" with an explanation as to why they may not be used.

("C")	"Confidential set/remove"
("D")	"Delivery basket set/remove"
("P")	"Priority/Normal message"
("R")	"Confirm receipt set/remove"
("S")	"edit Subject"
("T")	"edit Text"
("V")	"Vaporize date set/remove"
("X")	"Closed message set/remove"

- **OPTMSG^XMSEC2(XMDUZ,XMK,XMZ,.XMIM,.XMINSTR,.XMIU,.XMOPT)**

Determines what the user may do with the message. Some input parameters are set by calling INMSG1 and INMSG2^XMUTIL2. It does *not* set XMERR and ^TMP("XMERR", \$J).

Input Parameters:

XMDUZ	DUZ of the user who is accessing the message.
XMK	Basket IEN where the message is located: 0—if not in basket.
XMZ	Message IEN in the MESSAGE file (#3.9).
.XMIM	Message information, set by INMSG1^XMUTIL2: ("FROM") Who sent the message.
.XMINSTR	Set by INMSG2^XMUTIL2 "FLAGS", "TYPE", "VAPOR", "RCPT BSKT", "SCR HINT"
.XMIU	User information, as related to the message: ("ORIGN8") Did the user send the message? Set by INMSG2^XMUTIL2. ("IEN") User IEN in message RECIPIENT multiple, set by INMSG1^XMUTIL2.

Output Parameters:

.XMOPT	<p>Commands which the user may use are a subset of the following:</p> <p>If the user may not use them, they will have subnode(s) under "?" with an explanation as to why they may not be used.</p> <p>For example, if the message is Information Only, then XMOPT("R","?")="Only the sender may Reply to an 'Information only' message."</p> <p>If there were n lines of text, then XMOPT("R","?",1), ... XMOPT("R","?",n-1), and XMOPT("R","?") would be set.</p> <p>("A") "Answer" ("AA") "Access Attachments" ("B") "Backup" ("C") "Copy" ("D") "Delete" ("E") "Edit" ("F") "Forward" ("I") "Ignore" ("IN") "Information Only set/remove"</p>
--------	---

("H")	"Headerless Print"
("K")	"Priority replies set/remove"
("L")	"Later"
("N")	"New"
("P")	"Print"
("Q")	"Query"
("QR")	"Query Recipients"
("QD")	"Query Detailed"
("QN")	"Query Network"
("R")	"Reply"
("S")	"Save"
("T")	"Terminate"
("V")	"Vaporize date edit"
("W")	"Write"
("X")	"Xtract KIDS/PackMan"

23. Message and Mailbox Utility Functions and Routines—^XMXUTIL

- **\$\$BMSGCT^XMXUTIL(XMDUZ,XMK)**

Returns the number of messages in a user's basket.

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

- **\$\$BNMSGCT^XMXUTIL(XMDUZ,XMK)**

Returns the number of new messages in a user's basket.

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

- **\$\$BSKTNAME^XMXUTIL(XMDUZ,XMK)**

Returns the name of a user's basket.

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

- **\$\$NAME^XMXUTIL(XMDUZ,XMINFO)**

Returns the name of the user by looking up XMDUZ in the NEW PERSON file (#200). Optionally, it may also return the user's Title and/or Institution. If XMDUZ is not numeric, it returns XMDUZ.

Input Parameters:

XMDUZ	User DUZ.
XMINFO	(optional) If the variables XMV("SHOW INST") and XMV("SHOW TITL") indicate that the user's institution and/or title are desired, should that information be returned, too? 0—no (default) 1—yes

- **\$\$NETNAME^XMXUTIL(XMDUZ)**

Returns network name of user, including @site name.

Input Parameters:

XMDUZ	User DUZ or any string.
-------	-------------------------

- **\$\$NEWS^XMXUTIL(XMDUZ,XMTEST)**

Returns information on new messages in a user's mailbox. This function returns much the same information as the routine QMBOX^XMXAPI. It returns the following:

- 1—If XMDUZ is not a valid user.
- 0—If the user has no new messages.

Otherwise, it returns the following ^-delimited string:

- Piece 1: Number of new messages in the mailbox.
- Piece 2: Does the user have new priority mail?
0—no
1—yes
- Piece 3: Number of new messages in the "IN" basket.
- Piece 4: Date/time (in VA FileMan format) that the last message was received.
- Piece 5: Have there been any new messages since the last time this routine was called?
0—no
1—yes

Input Parameters:

XMDUZ User DUZ.

XMTEST (optional) Is this a test?
 0—no
 1—yes (default)

If this is *not* a test, then the LAST NEW MSG NOTIFY DATE/TIME field (#1.12) in the MAILBOX file (#3.7) may be updated for this user.

- **\$\$TMSGCT^XMXUTIL(XMDUZ)**

Returns the total number of messages in a user's mailbox.

Input Parameters:

XMDUZ User DUZ.

- **\$\$TNMSGCT^XMXUTIL(XMDUZ)**

Returns the total number of new messages in a user's mailbox.

Input Parameters:

XMDUZ User DUZ.

- **KVAPOR^XMXUTIL(XMDUZ,XMK,XMZ,XMVAPOR,.XMIU)**

Sets/Removes a message vaporize date in a user's basket.

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

XMZ Message IEN in the MESSAGE file (#3.9).

XMVAPOR Date/time (in VA FileMan format) to delete this message from this user's basket:
 ="@" (At-sign, Shift-2 key on most keyboards) to remove the vaporize date.

Output Parameters:

.XMIU User information, as related to the message:
 ("KVAPOR") Set to XMVAPOR or KILLED if XMVAPOR="@"

- **LASTACC^XMXUTIL(XMDUZ,XMK,XMZ,XMRESP,.XMIM,.XMINSTR,.XMIU,.XMCONFRM)**

Record that the user has read the message. This routine needs to be called only by those applications that display, using their own routines, messages and responses to the user. This routine sets the first and last times that the user has read the message. It records the last response that the user has read. If the user is a surrogate, it records the surrogate was the last reader. If MailMan had set a vaporize date for the message in the user's basket (because the user hadn't accessed it in a while), then that vaporize date is deleted. It also sends a confirmation message to the sender, if one was requested, the first time the user reads the message.

Input Parameters:

XMDUZ	User DUZ.
XMK	Basket IEN.
XMZ	Message IEN in the MESSAGE file (#3.9).
XMRESP	Last response read by the user this time.
.XMIM	Message information, set by INMSG1^XMXUTIL2: ("SUBJ") Subject. ("FROM") Sender.
.XMINSTR	More message information, set by INMSG2^XMXUTIL2: ("FLAGS") Special instructions (here, we are interested in whether "FLAGS"["R"—confirm receipt requested).
.XMIU	User information, as related to the message: ("IEN") IEN of user record in message RECIPIENT multiple, set by INMSG1^XMXUTIL2. ("RESP") Last response read by the user, initially set by INMSG1^XMXUTIL2 or INRESPTS^XMXUTIL2.

Output Parameters:

.XMIU	User information, as related to the message: ("RESP") If XMRESP is greater than XMIU("RESP"), then XMIU("RESP") is set to XMRESP.
.XMCONFRM	Was a confirmation message sent to the message sender? 0—no 1—yes

- **MAKENEW^XMXUTIL(XMDUZ,XMLK,XMLZ,XMLLOCKIT)**

Makes a message new and updates the new message counts.

Input Parameters:

XMDUZ	User DUZ.
XMLK	Basket IEN.
XMLZ	Message IEN in the MESSAGE file (#3.9).
XMLLOCKIT	(optional) Should MailMan take care of locking and unlocking the ^XMB(3.7,XMDUZ global? 0—no (default) 1—yes



The locking must be done to ensure the integrity of the new message counts. If MailMan doesn't do it, then the calling application must.

Output Parameters:

None.

- **NONEW^XMXUTIL(XMDUZ,XMLK,XMLZ,XMLLOCKIT)**

Makes a message *not* new and updates the new message counts.

Input Parameters:

XMDUZ	User DUZ.
XMLK	Basket IEN.
XMLZ	Message IEN in the MESSAGE file (#3.9).
XMLLOCKIT	(optional) Should MailMan take care of locking and unlocking the ^XMB(3.7,XMDUZ global? 0—no (default) 1—yes



The locking must be done to ensure the integrity of the new message counts. If MailMan doesn't do it, then the calling application must.

Output Parameters:

None.

- **PAGE^XMXUTIL(.XMABORT)**

Displays to the user: "Enter RETURN to continue or '^' to exit:" and waits until the user presses a key. Sets up and uses the standard VA FileMan call to do this.

Input Parameters:

None.

Output Parameters:

.XMABORT Did the user choose to exit? (0=no; 1=yes)

- **WAIT^XMXUTIL**

Displays to the user: "Press RETURN to continue:" and waits until the user presses a key. Sets up and uses the standard VA FileMan call to do this.

Input Parameters:

None.

Output Parameters:

None.

24. Date and String Utility Functions and Routines— ^XMXUTIL1

- **\$\$CONVERT^XMXUTIL1(X,XMTIME)**

Given an Internet DATE/TIME string, it returns the VA FileMan DATE/TIME. If the Internet DATE/TIME string *cannot* be understood, it returns -1.

Input Parameters:

X Internet DATE/TIME.

XMTIME (optional) Should the time also be converted?
0—no (default)
1—yes
If no, it returns the VA FileMan date only.

- **\$\$CTRL^XMXUTIL1(XMSTRING)**

Strip control characters from a string.

Input Parameters:

XMSTRING The string.

- **\$\$DECODEUP^XMXUTIL1(XMSTRING)**

Change all ~U~ to ^ in a string.

Input Parameters:

XMSTRING The string.

- **\$\$ENCODEUP^XMXUTIL1(XMSTRING)**

Change all ^ to ~U~ in a string.

Input Parameters:

XMSTRING The string.

- **\$\$GMTDIFF^XMXUTIL1(XMZONE)**

Given the time zone, it returns +-**hhmm** difference from Greenwich Mean Time (GMT). If there's no record of the time zone, it returns the null string.

Input Parameters:

XMZONE The 3-character time zone.

- **\$\$INDT^XMXUTIL1(XMDT)**

Given the VA FileMan DATE/TIME, it returns the Internet DATE/TIME string:

dd mm yyy hh:mm:ss +-hhmm (time zone)

Input Parameters:

XMDT VA FileMan DATE/TIME.

- **\$\$MAXBLANK^XMXUTIL1(XMSTRING)**

Reduce all three or more consecutive blanks in a string to two.

Input Parameters:

XMSTRING The string.

- **\$\$MELD^XMXUTIL1(XMSTRING,XMNUMBER,XMLEN)**

Combine a string and a number to form a new string of a given length. The string will be right justified; the number left-justified, with at least two blanks separating the string and number. The string will be truncated, if necessary.

For example:

`$$MELD^XMXUTIL1("Lotus blossom",123,10)` returns "Lotus 123"

Input Parameters:

XMSTRING The string.

XMNUMBER (optional) The number.

XMLEN The length of the new string to be formed.

- **\$\$MMDT^XMXUTIL1(XMDT)**

Given the VA FileMan DATE/TIME, it returns the following string:

dd mmm yy hh:mm

Input Parameters:

XMDT VA FileMan DATE/TIME.

- **\$\$SCRUB^XMXUTIL1(XMSTRING)**

Strip control characters and leading/trailing blanks from a string.

Input Parameters:

XMSTRING The string.

- **\$\$STRIP^XMXUTIL1(XMSTRING)**

Strip leading/trailing blanks from a string.

Input Parameters:

XMSTRING The string.

- **\$\$TIMEDIFF^XMXUTIL1(XMDIFF)**

Given the decimal time difference (between time zones), it returns **+hhmm** (e.g., **-2.5** equates to **-0230**).

Input Parameters:

XMDIFF Decimal time difference.

- **\$\$TSTAMP^XMXUTIL1()**

Return a timestamp (**\$H** expressed in seconds).

Input Parameters:

None.

- **ZONEDIFF^XMXUTIL1(XMYT,.XMHH,.XMMM)**

Given the time zone (or time difference +-**hhmm** from Greenwich Mean Time [GMT]), it returns the number of hours and minutes difference between that and the local time zone.

Input Parameters:

XMYT Time zone (3-character or +-**hhmm** from GMT).

Output Parameters:

.XMHH Number of hours time difference.

.XMMM Additional number of minutes time difference.

25. Message Information Functions and Routines— ^XMXUTIL2 & ^XMXUTIL3

These functions and routines retrieve all kinds of information about a message:

- Information that can be displayed.
- Information that can be used to determine what may (and may not) be done with the message.

^XMXUTIL2

- **\$\$BSKT^XMXUTIL2(XMDUZ,XMZ,XMNAME)**

Returns which basket a message is in for a user. It returns the following:

0—Not in a basket for this user.

Number—It's in this basket IEN for the user. (XMNAME=0)

Number^name—It's in this basket IEN of this name for the user. (XMNAME=1)

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

XMNAME (optional) Return the basket name, too?

0—no (default)

1—yes

- **\$\$DATE^XMXUTIL2(XMZREC,XMTIME)**

Returns the message sent date. It is returned in external format:

DD MMM YY HH:MM



Compare to \$\$ZDATE^XMXUTIL2 described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9,XMZ,0).

XMTIME (optional) Return the time, also?

0—no, date only

1—yes, date and time (default)

- **\$\$FROM^XMXUTIL2(XMZREC)**

Returns the message From information. It is returned in external format.



Compare to \$\$ZFROM^XMXUTIL2 described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$KSEQN^XMXUTIL2(XMDUZ, XMK, XMZ)**

Returns the sequence number for a message in this user's basket.

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$LINE^XMXUTIL2(XMZ)**

Returns the number of lines in the text of a message.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$NEW^XMXUTIL2(XMDUZ, XMK, XMZ)**

Returns a value indicating whether or not a message is new for this user in this basket (0=no; 1=yes).

Input Parameters:

XMDUZ User DUZ.

XMK Basket IEN.

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$PRI^XMXUTIL2(XMZREC)**

Returns a value indicating whether the message is priority or not (0=no;1=yes).



Compare to \$\$ZPRI^XMXUTIL2 described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9,XMZ,0).

- **\$\$QRESP^XMXUTIL2(XMZ,XMZREC,XMWHICH)**

Determines whether a message is a response or not. It returns the following:

0—Message XMZ is not a response.

IEN^number—It's a response to this message IEN in the MESSAGE file (#3.9). And it is response <number> to that message. This 2nd piece is only returned if XMWHICH=1.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMZREC (optional) Zero node of the message: ^XMB(3.9,XMZ,0).

XMWHICH (optional) If it is a response to a message, do you want to know which number response?

0—no (default)

1—yes

- **\$\$RESP^XMXUTIL2(XMZ)**

Returns the number of responses to a message.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$SUBJ^XMXUTIL2(XMZREC)**

Returns the message subject. It is returned in external format.



Compare to \$\$ZSUBJ^XMXUTIL2 described below.

Input Parameters:

XMZREC Zero node of the message: ^XMB(3.9, XMZ, 0).

- **\$\$ZDATE^XMXUTIL2(XMZ, XMTIME)**

Returns the message sent date. It is returned in external format:

DD MMM YY HH:MM.



Compare to \$\$DATE^XMXUTIL2 described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

XMTIME (optional) Return the time, also?

0—no, date only

1—yes, date and time (default)

- **\$\$ZFROM^XMXUTIL2(XMZ)**

Returns the message From. It is returned in external format.



Compare to \$\$FROM^XMXUTIL2 described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZNODE^XMXUTIL2(XMZ)**

Returns the message zero node: ^XMB(3.9, XMZ, 0).

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZPRI^XMXUTIL2(XMZ)**

Returns a value indicating whether the message is priority or not (0=no;1=yes).



Compare to \$\$PRI^XMXUTIL2 described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZREAD^XMXUTIL2(XMDUZ, XMZ)**

Returns the number of responses to a message this user has read. It returns the following:

 null—User has not read the message at all.

 0—User has read the original message only.

 Number—User has read through this response.

Input Parameters:

XMDUZ User DUZ.

XMZ Message IEN in the MESSAGE file (#3.9).

- **\$\$ZSUBJ^XMXUTIL2(XMZ)**

Returns the message subject. It is returned in external format.



Compare to \$\$SUBJ^XMXUTIL2 described above.

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

- **INMSG^XMXUTIL2(XMDUZ,XMK,XMZ,XMZREC,XMFLAGS,.XMIM,.XMINSTR,.XMIU)**

Message information.



This routine should only be called for messages, not for responses. This routine calls both INMSG1^XMXUTIL2 and INMSG2^XMXUTIL2. It also returns additional information.

Input Parameters:

- | | |
|---------|--|
| XMDUZ | User DUZ. |
| XMK | Basket IEN.

(Set XMK=0, if the message is <i>not</i> in a basket or if you are <i>not</i> interested in variables XMIU("KVAPOR") and XMIU("NEW").) |
| XMZ | Message IEN in the MESSAGE file (#3.9). |
| XMZREC | (optional) Zero node of the message: ^XMB(3.9,XMZ,0). |
| XMFLAGS | (optional) Used to control output:

I—Internal values only. (Default is internal values, and, where it makes sense, to set variables with other values, too.)

F—Set variable with internal VA FileMan date format. (Default is external MailMan date format.) "F" is ignored if XMFLAGS contains "I". |

Output Parameters:

- | | |
|----------|---|
| .XMIM | Message information (KILLED first). For a description of this parameter, please refer to the definition of XMIM for INMSG1^XMXUTIL2 (described below):

"SUBJ", "ENV FROM", "FROM", "FROM DUZ", "FROM NAME", "DATE", "DATE FM", "DATE MM", "SENDER", "SENDER DUZ", "SENDER NAME", "LINES", "RESPS", "RECIPS", "CRE8", "CRE8 MM" |
| .XMINSTR | Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:

"FLAGS", "TYPE", "VAPOR", "RCPT BSKT", "SCR HINT" |

.XMIU

User information, as related to the message. For a description of this parameter, please refer to the definition of XMIU for INMSG1^XMXUTIL2 and INMSG2^XMXUTIL2 (described below):

"IEN", "RESP", "ORIGN8"

("KVAPOR") DATE/TIME (in VA FileMan format) to delete this message from this user's basket. (Set only if applicable.)

("NEW") Is message new?

0—no

1—yes

2—yes, and priority, too.

The following table compares the variables returned by \$\$HDR^XMGAPI2 and INMSG^XMXUTIL2:

\$\$HDR^XMGAPI2	INMSG^XMXUTIL2
L("BLOBCNT")	N/A
L("BROADCAST")	N/A (use \$\$BCAST^XMXSEC)
L("BSKT IEN")	N/A (use \$\$BSKT^XMXUTIL2)
L("BSKT")	N/A (use \$\$BSKT^XMXUTIL2)
L("DATE FM")	XMIM("DATE FM")—If XMFLAGS["F" and XMFLAGS["I".
L("DATE")	XMIM("DATE")—Internal.
L("LINES")	XMIM("LINES")
L("NEW")	XMIU("NEW")
L("PXMZ")	N/A
L("RRCV")	XMIM("RESPS")
L("RRED")	XMIU("RESP")
L("RSP",i)	N/A (use INRESP^XMXUTIL2 to get response information)
L("SENDER DUZ")	XMIM("FROM DUZ")—If XMFLAGS["I".
L("SENDER")	XMIM("FROM NAME")—If XMFLAGS["I".
L("SUBJ")	XMIM("SUBJ")
L("SURROG")	XMIM("SENDER NAME")—If XMFLAGS["I".
L("TYPE")	XMINSTR("TYPE")
L("XMZ")	XMIM("XMZ")
N/A	XMIM("CRE8")—Local create date.
N/A	XMIM("CRE8 MM")—MailMan external formatted date, if XMFLAGS["F" or "I".

Table 4: Comparison of Variables Returned By \$\$HDR^XMGAPI2 and INMSG^XMXUTIL2

Table 4 (continued):

\$\$HDR^XMGAPI2	INMSG^XMXUTIL2
N/A	XMIM("DATE MM")—MailMan external formatted date, if XMFLAGS["F" or "I".
N/A	XMIM("ENV FROM")—Message envelope "MAIL FROM:"
N/A	XMIM("FROM")—Internal.
N/A	XMIM("RECIPS")—Number of recipients.
N/A	XMIM("SENDER DUZ")—If XMFLAGS["I".
N/A	XMIM("SENDER")—Internal.
N/A	XMINSTR("FLAGS")—Closed, Confidential, Information Only, Priority, Confirmation requested, Priority responses.
N/A	XMINSTR("RCPT BSKT")—Delivery basket.
N/A	XMINSTR("SCR HINT")—Scramble hint.
N/A	XMINSTR("VAPOR")—Vaporize date of message.
N/A	XMIU("IEN")—User's IEN in RECIPIENT multiple.
N/A	XMIU("KVAPOR")—Vaporize date of message in user's basket.
N/A	XMIU("ORIGN8")—Did user send message?

Table 4: Comparison of Variables Returned By \$\$HDR^XMGAPI2 and INMSG^XMXUTIL2 (continued)

- **INMSG1^XMXUTIL2(XMDUZ,XMZ,XMZREC,XMFLAGS,.XMIM,.XMIU)**

Message information, Part 1.



This routine should only be called for messages, not for responses. It calls routine INRESPS^XMXUTIL2.

Input Parameters:

- | | |
|---------|--|
| XMDUZ | User DUZ. |
| XMZ | Message IEN in the MESSAGE file (#3.9). |
| XMZREC | (optional) Zero node of the message: ^XMB(3.9,XMZ,0). |
| XMFLAGS | (optional) Used to control setting of output variables:
I—Internal values only (Default is internal values, and, where it makes sense, to set variables with other values, too.)
F—Set variable with internal VA FileMan date format. (Default is external MailMan date format.) "F" is ignored if XMFLAGS contains "I". |

Output Parameters:

- | | |
|---------------|--|
| .XMIM | Message information (KILLED first): |
| ("XMZ") | Message IEN in the MESSAGE file (#3.9). |
| ("SUBJ") | Subject of message (all ~U~ translated to ^). |
| ("ENV FROM") | "MAIL FROM:", as stated in the message envelope on a message that was received from a remote site.
(<i>Not set, if it doesn't exist.</i>) |
| ("FROM") | Who sent the message (internal). |
| ("FROM DUZ") | DUZ of person who sent the message (if applicable).
(<i>Not set if XMFLAGS contains "I".</i>) |
| ("FROM NAME") | Name of person who sent the message.
(<i>Not set if XMFLAGS contains "I".</i>) |
| ("DATE") | When the message was sent (internal). |
| ("DATE FM") | VA FileMan date (-1, if error).
(Set if XMFLAGS contains "F". <i>Not set if XMFLAGS contains "I".</i>) |

("DATE MM")	External MailMan format: dd mmm yy hh:mm (Internet date, if error.) (<i>Not set, if XMFLAGS contains "I" or "F".</i>)
("CRE8")	Local create date (in VA FileMan format).
("CRE8 MM")	External MailMan format: dd mmm yy hh:mm (<i>Not set, if XMFLAGS contains "I" or "F".</i>)
("SENDER")	Who really sent the message (if applicable).
("SENDER DUZ")	DUZ of person who really sent the message (if applicable). (<i>Not set, if XMFLAGS contains "I".</i>)
("SENDER NAME")	Name of person who really sent the message (if applicable). (<i>Not set, if XMFLAGS contains "I".</i>)
("LINES")	How many lines are in the message.
("RESPS")	How many responses does the message have.
.XMIU	User information, as related to the message (KILLED first).
("IEN")	IEN of XMDUZ in the message's RECIPIENT multiple.
("RESP")	Number of the last response that the user has read.

- **INMSG2^XMXUTIL2(XMDUZ,XMZ,XMZREC,.,XMIM,.,XMINSTR,.,XMIU)**

Message information, Part 2.



This routine should only be called for messages, not for responses.

Input Parameters:

XMDUZ	User DUZ.
XMZ	Message IEN in the MESSAGE file (#3.9).
XMZREC	(optional) Zero node of the message: ^XMB(3.9,XMZ,0).

Output Parameters:

- .XMIM Message information.
("RECIPS") Number of recipients of the message.
- .XMINSTR Special instructions on the message. For a description of this parameter, please refer to the "Parameter Definitions" list in Chapter 16 in this manual:
"FLAGS", "TYPE", "VAPOR", "RCPT BSKT", "SCR HINT"
- .XMIU User information, as related to the message:
("ORIGN8") Did the user send the message?
0—no
1—yes
(results from a call to \$\$ORIGIN8R^XMXSEC)

- **INRESP^XMXUTIL2(XMZ,XMWHICH,XMFLAGS,,XMIR)**

Response information.



This routine should only be called for responses, not for messages.

Input Parameters:

- XMDUZ User DUZ.
- XMZ Message IEN in the MESSAGE file (#3.9).
- XMWHICH The number of the response for which to get the information.
- XMFLAGS (optional) Used to control output:
I—Internal values only. (Default is internal values, and, where it makes sense, to set variables with other values, too.)
F—Set variable with internal VA FileMan date format. (Default is external MailMan date format.) "F" is ignored if XMFLAGS contains "I".

Output Parameters:

- .XMIR Response information (KILLed first). For a description of this parameter, please refer to the definition of XMIM for INMSG1^XMXUTIL2 (previously described):
"XMZ", "SUBJ", "ENV FROM", "FROM", "FROM DUZ", "FROM NAME", "DATE", "DATE FM", "DATE MM", "SENDER", "SENDER DUZ", "SENDER NAME", "LINES"

- **INRESPS^XMXUTIL2(XMZ,.XMIM,.XMIU)**

How many responses? What has the user read?

Input Parameters:

XMZ Message IEN in the MESSAGE file (#3.9).

.XMIU("IEN") Set by INMSG1^XMXUTIL2.

Output Parameters:

.XMIM("RESPS") Number of responses for a message.

.XMIU("RESP") Number of the last response that the user has read.

^XMXUTIL3

- **Q^XMXUTIL3(XMZ,XMFLAGS,XMAMT,.XMSTART,XMFIND,XMTROOT)**

Get a list of the addressees of this message. Optionally, find addressees, which match a string. Gets a list (similar in format to that produced by LIST^DIC) of the requested addressees.

Input Parameters:

XMZ	Message IEN in the MESSAGE file (#3.9).
XMFLAGS	(Reserved for future use.)
XMAMT	(optional) How many? Number—Get this many. *—Get all (default).
.XMSTART	(optional) Used to start the Lister. The Lister keeps it updated from call to call. ("IEN") Start <i>after</i> this addressee IEN. Continues from there, with each successive call, to the end. (Default is to start at the beginning.)
XMFIND	(optional) Find addressees that match the string. (VA FileMan's FIND^DIC is used.) If XMFIND is supplied, then XMAMT and XMSTART are ignored; a complete list is always returned.
XMTROOT	Target root (closed) to receive the message list. (Default is ^TMP("XMLIST", \$J).)

Output Parameters:

.XMSTART	(As defined above.)
XMTROOT	Fields returned under XMTROOT for each addressee: "TO NAME" Addressee name. "TYPE" Addressee type (if present): I—Information Only C—cc (Carbon Copy)

- **QD^XMXUTIL3(XMZ,XMFLAGS,XMAMT,.XMSTART,XMFIND,XMTROOT)**

Get a list of the recipients of this message. Optionally, find recipients that match a string. Gets a list (similar in format to that produced by LIST^DIC) of the requested recipients.

Input Parameters:

XMZ	Message IEN in the MESSAGE file (#3.9).
XMFLAGS	(Reserved for future use.)
XMAMT	(optional) How many? Number—Get this many *—Get all (default)
.XMSTART	(optional) Used to start the Lister going. The Lister will keep it updated from call to call. ("IEN") Start <i>after</i> this recipient IEN. Continues from there, with each successive call, to the end. (Default is to start at the beginning.)
XMFIND	(optional) Find recipients that match the string. (VA FileMan's FIND^DIC is used.) If XMFIND is supplied, then XMAMT and XMSTART are ignored; a complete list is always returned.
XMTROOT	Target root (closed) to receive the message list. (Default is ^TMP("XMLIST", \$J).)

Output Parameters:

.XMSTART	(As defined above.)
XMTROOT	Fields returned under XMTROOT for each recipient: "TO" Recipient .01 field (could be DUZ or text). "TO NAME" Recipient name. "TO ID" ID of recipient: L—Local user F—Fax R—Remote S—Server D—Device *—Broadcast

"TYPE"	(if present) Recipient type: I—Information Only C—cc (Carbon Copy)
"FWD BY DUZ"	(if present) DUZ of the person who forwarded to this recipient.
"FWD BY"	(if present) Name of the person, possibly followed by, in parentheses, the name of the surrogate of the person, who forwarded to this recipient.
"FWD ON"	Date that message was forwarded to this recipient in MailMan format: dd mmm yy hh:mm

Depending on "TO ID", the following fields are also returned:

➤ **"TO ID"="L"—Local User:**

"TO DUZ"	DUZ of the local recipient.
"RESP"	(if present) Number of the last response read (zero equals original message).
"LREAD"	(if present) DATE/TIME (in VA FileMan format) the message was last read.
"LREAD MM"	(if present) DATE/TIME (in MailMan format) the message was last read.
"FREAD"	(if present) DATE/TIME (in VA FileMan format) the message was first read.
"FREAD MM"	(if present) DATE/TIME (in MailMan format) the message was first read.
"COPY"	(if present) DATE/TIME (in VA FileMan format) the message was last copied.
"COPY MM"	(if present) DATE/TIME (in MailMan format) the message was last copied.
"TERM"	(if present) DATE/TIME (in VA FileMan format) the message was terminated.
"TERM MM"	(if present) DATE/TIME (in MailMan format) the message was terminated.
"SURR"	(if present) Name of the surrogate who last read the message.

➤ **"TO ID"="*"—Broadcast:**

No additional fields.

➤ **"TO ID"="F"—Fax:**

"FDATE"	(if present) DATE/TIME (in VA FileMan format) the message was passed to the Fax software.
"FDATE MM"	(if present) DATE/TIME (in MailMan format) the message was passed to the Fax software.
"STATUS"	(if present) Status of the fax (present before the message is passed to the Fax software).
"FAX IEN"	(if present) IEN (in FAX ROLODEX file) of the fax recipient (present before the message is passed to the Fax software).
"ID"	(if present) Fax ID (present after the message is passed to the Fax software).

➤ **"TO ID"="R"—Remote:**

"XDATE"	(if present) DATE/TIME (in VA FileMan format) the transmission of the message began (present after transmission is complete).
"XDATE MM"	(if present) DATE/TIME (in MailMan format) the transmission of the message began (present after transmission is complete).
"STATUS"	(if present) Status of the message (present before and during transmission, and if error, afterward).
"ID"	(if present) Message ID (present after transmission is successfully completed).
"PATH"	(if present) IEN (in DOMAIN file [#4.2]) of the Domain to/via which the message will be sent (present before and during transmission).
"PATH NAME"	(if present) Name of the Domain to/via which the message will be sent (present before and during transmission).
"SECS"	(if present) Duration of the transmission (in seconds, present after transmission is complete).

➤ **"TO ID"="D"—Device or "S"—Server:**

"SDATE"	(if present) DATE/TIME (in VA FileMan format) each time the STATUS changes, once a task starts dealing with the message.
"SDATE MM"	(if present) DATE/TIME (in MailMan format) each time the STATUS changes, once a task starts dealing with the message.
"STATUS"	(if present) Status of the message (usually present before, during, and after sending).

- **QL^XMXUTIL3(XMZ,XMFLAGS,XMAMT,.XMSTART,XMFIND,XMTROOT)**

Get a list of the "latered" addressees of this message. Optionally, find the "latered" addressees that match a string. Gets a list (similar in format to that produced by LIST^DIC) of the requested "latered" addressees.

Input Parameters:

XMZ	Message IEN in the MESSAGE file (#3.9).
XMFLAGS	(Reserved for future use.)
XMAMT	(optional) How many? Number—Get this many. *—Get all (default).
.XMSTART	(optional) Used to start the Lister. The Lister will keep it updated from call to call. ("IEN") Start <i>after</i> this latered addressee IEN. Continues from there, with each successive call, to the end. (Default is to start at the beginning.)
XMFIND	(optional) Find the "latered" addressees that match the string. (VA FileMan's FIND^DIC is used.) If XMFIND is supplied, then XMAMT and XMSTART are ignored; a complete list is always returned.
XMTROOT	Target root (closed) to receive the message list. (Default is ^TMP("XMLIST", \$J).)

Output Parameters:

.XMSTART	(As defined above.)
XMTROOT	Fields returned under XMTROOT for each "latered" addressee: "TO NAME" Latered addressee name. "TYPE" (if present) Addressee type: I—Information Only C—cc (Carbon Copy) "BY DUZ" DUZ of the person who "latered." "BY NAME" Name of the person who "latered." "WHEN" When will the message be delivered (in VA FileMan format). "WHEN MM" When will the message be delivered (in MailMan format): dd mmm yy hh:mm

- **QN^XMXUTIL3(XMZ,XMFLAGS,XMAMT,.,XMSTART,XMTROOT)**

Get the network header records from a message that originated at a remote site. Gets a list (similar in format to that produced by LIST^DIC) of the message's network header records.

Input Parameters:

XMZ	Message IEN in the MESSAGE file (#3.9).
XMFLAGS	(Reserved for future use.)
XMAMT	(optional) How many? Number—Get this many *—Get all (default)
.XMSTART	(optional) Used to start the Lister going. The Lister will keep it updated from call to call. ("IEN") Start <i>after</i> this line IEN. Continues from there, with each successive call, to the end.(Default is to start at the beginning.)
XMTROOT	Target root (closed) to receive the message list. (Default is ^TMP("XMLIST", \$J).)

Output Parameters:

.XMSTART	(As defined above.)
----------	---------------------

Example:

```

>D QN^XMXUTIL3(978437)
>D ^%G

Global ^TMP("XMLIST", $J
      TMP("XMLIST", $J
^TMP("XMLIST", 541073053, 0) = 12^^^0
^TMP("XMLIST", 541073053, 1) = Received: from ISC-SF.VA.GOV by
MAILMAN.ISC-SF.VA.GOV (MailMan/7.1 Turn Around) id 978437 ; 1 May 1998
06:38:29 -0700 (PDT)
^TMP("XMLIST", 541073053, 2) = Received: from FORUM.VA.GOV by ISC-
SF.VA.GOV (MailMan/7.1 TCP/IP-MAILMAN) id 1204177 ; 11 Mar 1998
09:25:25 -0800 (PST)
^TMP("XMLIST", 541073053, 3) = Subject:Released DI*21*43 SEQ #39
^TMP("XMLIST", 541073053, 4) = Date:11 Mar 98 12:22 EST
^TMP("XMLIST", 541073053, 5) = Message-ID:<26463552@FORUM.VA.GOV>
^TMP("XMLIST", 541073053, 6) = From:<National Patch Module@FORUM.VA.GOV>
^TMP("XMLIST", 541073053, 7) = To: PUCE.FIL@FORUM.VA.GOV,
G.PATCH@FORUM.VA.GOV, G.SUPPORT@FORUM.VA.GOV, G.SUPPORT@ISC-
ALBANY.VA.GOV,
^TMP("XMLIST", 541073053, 8) =      G.SUPPORT@ISC-BIRM.VA.GOV,
G.SUPPORT@ISC-CHICAGO.VA.GOV, G.SUPPORT@ISC-DALLAS.VA.GOV,
^TMP("XMLIST", 541073053, 9) =      G.SUPPORT@ISC-SF.VA.GOV,
G.SUPPORT@ISC-SLC.VA.GOV, S.A1AE SERVER VERIFIED@FORUM.VA.GOV,
^TMP("XMLIST", 541073053, 10) =      PARKS.RICHARD@FORUM.VA.GOV,
NAPOLI.GERALD@FORUM.VA.GOV, HODGES.BRYAN@FORUM.VA.GOV,
^TMP("XMLIST", 541073053, 11) =      G.CSNATHD@FORUM.VA.GOV
^TMP("XMLIST", 541073053, 12) =
Global ^

```


Glossary

.001 FIELD	A field containing the internal entry number of the record.
.01 FIELD	The one field that must be present for every file and file entry. It is also called the NAME field. At a file's creation the .01 field is given the label NAME. This label can be changed.
APPLICATION PROGRAMMER INTERFACE (API)	Programmer calls provided by MailMan for use by application programmers. APIs allow programmers to carry out standard computing activities without needing to duplicate MailMan utilities in their own packages. APIs also further DBA goals of system integration by channeling activities, such as adding new users, through a limited number of callable entry points.
ASCII	American Standard Code for Information Interchange. A standardized coding scheme that assigns numeric values to letters, numbers, punctuation marks, and other characters to enable computer systems to exchange information.
AT-SIGN ("@")	A VA FileMan security Access Code that gives the user programmer-level access to files and to VA FileMan's developer features. See Programmer Access. Also, the character "@" (i.e., At-sign, Shift-2 key on most keyboards) is used at VA FileMan field prompts to delete data.
BLOB	Binary Large Objects.
BOOLEAN EXPRESSION	A logical comparison between values yielding a true or false result. In M, zero means false and non-zero (often one) means true.
BULLETIN	Electronic mail messages that are automatically delivered by MailMan under certain conditions. For example, a bulletin can be set up to fire when database changes occur, such as adding a record to the file of users. Bulletins are fired by bulletin-type cross-references.

CROSS-REFERENCE	<p>An attribute of a field or a file that identifies an action that should take place when the value of a field is changed. Often, the action is the placement of the field's value into an index. A Traditional cross-reference is defined with a specific field. A New-Style cross-reference is a file attribute and can be composed of one or more fields. New-Style cross-references are stored in the INDEX file (#.11).</p> <p>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 (SSN), and bed number. When asked for a patient, the user can then respond with the patient's name, SSN, 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.</p>
DATA DICTIONARY	<p>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.</p>
DATA DICTIONARY ACCESS	<p>A user's authorization to write, update, and/or edit the data definition for a computer file, also known as DD Access.</p>
DBIA	<p>Database Integration Agreement, a formal understanding between two or more VISTA packages which describes how data is shared or how packages interact. The DBA maintains a list of DBIAs between package developers allowing the use of internal entry points or other package-specific features that are not available to the general programming public.</p>
DOMAIN	<p>A site for sending and receiving mail.</p>
DOUBLE QUOTE (")	<p>A symbol used in front of a Common option's menu text or synonym to select it from the Common menu. For example, the four-character string "TBOX" selects the User's Toolbox Common option.</p>
FIELD	<p>In an entry, a specified area used to hold values. The specifications of each VA FileMan field are documented in the file's data dictionary.</p>
FIELD NUMBER	<p>The unique number used to identify a field in a file. A field can be referenced by "#" followed by the field number.</p>
FREE TEXT	<p>The use of any combination of numbers, letters, and symbols when entering data.</p>

HELP PROMPT	The brief Help that is available at the field level when entering one or more question marks.												
INITIALIZATION	The process of setting variables in a program to their starting value.												
INPUT TRANSFORM	An executable string of M code that is used to check the validity of input and converts it into an internal form for storage.												
INTERNAL ENTRY NUMBER	The number used to identify an entry within a file. Every record has a unique internal entry number. Often abbreviated as IEN.												
IRM	Information Resource Management. A service at VA sites responsible for computer management and system security.												
KERNEL	A set of VISTA software routines that function as an intermediary between the host operating system and the VISTA application packages (e.g., Laboratory, Pharmacy, IFCAP, etc.). Kernel provides a standard and consistent user and programmer interface between application packages and the underlying M implementation. (VA FileMan and MailMan are self-contained to the extent that they can stand alone as verified packages.) Some of Kernel's components are listed below along with their associated namespace assignments: <table data-bbox="641 1035 1040 1234" style="margin-left: 40px;"> <tr> <td>KIDS</td> <td>XPD</td> </tr> <tr> <td>Menu Management</td> <td>XQ</td> </tr> <tr> <td>Tools</td> <td>XT</td> </tr> <tr> <td>Sign-on/Security</td> <td>XU</td> </tr> <tr> <td>Device Handling</td> <td>ZIS</td> </tr> <tr> <td>Task Management</td> <td>ZTM</td> </tr> </table>	KIDS	XPD	Menu Management	XQ	Tools	XT	Sign-on/Security	XU	Device Handling	ZIS	Task Management	ZTM
KIDS	XPD												
Menu Management	XQ												
Tools	XT												
Sign-on/Security	XU												
Device Handling	ZIS												
Task Management	ZTM												
KEY	The purpose of Security Keys is to set a layer of protection on the range of computing capabilities available with a particular software package. The availability of options is based on the level of system access granted to each user. Security keys are assigned to individual users. Keys allow access to options.												
KIDS	Kernel Installation and Distribution System.												
LOCAL	The system to which a user is currently signed on.												
LOOKUP	To find an entry in a file using a value for one of its fields.												
MESSAGE-ID	A message identifier which shows the message number and the domain name of the message.												
MIME	Multipurpose Internet Mail Extension. A standard that allows you to create structured message bodies.												

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, can be sent and received either locally or across the network.
MULTIPLE	A DATA TYPE that allows more than one value for a single entry. Also known as a Subfile.
MUMPS	Abbreviated as M. The American National Standards Institute (ANSI) computer language used by MailMan and throughout VISTA. The acronym MUMPS stands for M assachusetts G eneral H ospital U tility M ulti P rogramming S ystem.
NAME FIELD	The one field that must be present for every file and file entry. It is also called the .01 field. At a file's creation the .01 field is given the label NAME. This label can be changed.
NULL	Empty. A field or variable that has no value associated with it is null.
PATTERN MATCH	In M, an operator that compares the contents of a variable or literal to a specified pattern of characters or kinds of characters.
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.
POSTMASTER	The basket where message queues are stored. Also, the person who manages this basket for a particular site.
PROGRAMMER ACCESS	The ability to use VA FileMan features that are reserved for application developers. Referred to as "having the At-sign ('@')" because the At-sign is the DUZ(0) value that grants programmer access.
PROTOCOL	Code containing logic for opening and closing links, and for sending/receiving transmissions.
PURGE	A procedure used to delete messages or message pointers.
READ ACCESS	A user's authorization to read information stored in a computer file.
ROUTINE	A program or a sequence of instructions called by a program that may have some general or frequent use. M routines are groups of program lines that are saved, loaded, and called as a single unit via a specific name.

SERVER	An automatic mail reader for internal messages.
SURROGATE	A person who is authorized to read and/or send mail in another user's name.
TASKMAN	The Kernel module that schedules and processes background tasks (also called Task Manager).
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.
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 online computer routines written in the M language which 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.
VARIABLE	A symbol representing a value that changes during the execution of a routine.
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).

Appendix A—Message Server Protocol

Overview

A server is an option that is invoked when a mail message that has been addressed to it has been delivered. As an option, many of the parameters associated with the servers are embedded in the definition of the option.



For more information on servers, please refer to "Chapter 12: Servers" in "Part 2: Menu Manager" in the "Kernel V 8.0 Systems Manual." Options are also described in "Part 2: Menu Manager" in the "Kernel V 8.0 Systems Manual."

Servers may or may not receive data. The received data usually comes in the form of the text of the message being delivered to it, but the data may also be pointed to by the message and exist in the system, either because it was there in the beginning or because it arrived independently.

Servers can be addressed from a remote site. For example, a server on ALTOONA.MED.VA.GOV can receive a message addressed to it from WASHINGTON.MED.VA.GOV. In fact, this is very common. Because of this capability, an option that has been designated as a server has security features as parameters. Please be aware of these security parameters. Messages addressed to servers will *not* be scheduled, if security is *not* passed.

Filegrams work through the use of a server. Data is loaded into a mail message, addressed and when delivered, processed by the filegram server into the receiving database.

Servers are always invoked through tasks that are set up when the message is delivered into the system, locally or over the network. One of the options is to "Run Immediately." Then the task is scheduled to run "NOW."

However, tasks may not need to be scheduled at all, because the system manager has stated so in the entry for the server in the OPTION file (#19) or because of a problem.

Server Statuses

Server recipients are recorded in the recipient chain of a message and appear similarly to other users. MailMan enters statuses on its own as stages in the server process are reached. First, the message is marked as "Awaiting Server." This indicates that the message has been received and the option is a valid one. At this point, a task has been created to actually invoke MenuMan to schedule or perform the service (option) required.

The last status that MailMan sets is "Served," which means that MenuMan has been called successfully and MenuMan has either performed the task in the case of a server that runs immediately or that some other action has been taken.

At this point, a task could be scheduled to invoke the server, a message could be sent to indicate that the task exists and needs to be scheduled, or some other action that was required was performed. MenuMan has its own statuses that will be used.

\$\$SRVTIME^XMS1

This extrinsic function sets the status of recipients in a message.



For details on this function, please refer to Chapter 13 in this manual..

Addressing a Server

To address a server, precede the recipient name with "S." (e.g., S.XMECHO, this example sends a message to the MailMan Echo Tester server). "S." must be followed by an option name from the OPTION file (#19) in the Target Domain. If not, a "Recipient not Found" error will occur.

A "Recipient Ambiguous" error will occur, if there is more than one option whose name partially matches the name addressed.

For example, the District Registry server for admitting a new patient could be addressed as follows:

```
S.DGDI$TADMIT@SANFRANCISCO.MED.VA.GOV
```

The message is destined for the DGDISTADMIT option at San Francisco. Replies to this message would be from this same name.

Writing a Server Program

The server communicates with mail messages in specific ways. Code is used to interface the server to the message system. The code below returns the original message to the sender:

```
ECHO      ;
K XMY
S XMSUB=$E("Server echo of '_XMBSUB_'`",1,65)
S XMY(XMFROM)=",XMTXT="^XMB(3.9,"_XMZ_",2,"
D ^XMD
Q
```

In this example, the variable XMFROM contains the sender address and is supplied to the server when invoked. Other variables also exist upon invocation of the server.

Routine ^XMF1 is an example server program supplied with MailMan. ^XMF1 uses some of the other variables supplied to the server.

Execute variable XMREC to read a line of the message. XMER and XMRG are returned.

- XMER** This variable returns the execution status of XMREC. XMER<0, if there is no more message text to read. The value of XMER will be zero (0), if XMRG is being returned as non-null. XMRG, in that case, will have as its value the text of the next line of the message.
- XMRG** The value of XMRG will be the next line of message text. XMRG will always be defined, though it will be null when XMER<0.
- XMPOS** This variable contains the current position of the text returned in the variable XMGR. It is initialized if it is undefined, but should be KILLED by the server when it is finished "Reading" the message.

Here's another example of code, this time from XMF1:

```
S XMA=0
A;
X XMREC ; Receive a line
I $D(XMER) G Q;XMER<0 ; Check for end of message
S XMA=XMA+1 ; Increment local line count
S XMTEXT(XMA)=XMRG ; Set local array
G A ; Go back for another line
```

Double Serving Messages

On occasion, a system backup interrupts the transmission/receive process. It appears to result in the same message being served twice. The Audit Log for the OPTION file shows two messages with the same message number and subject, but with different Date/Times and Job Numbers.

To avoid this, application servers should be written such that they check for and avoid processing of the same message being delivered to any particular server. MailMan transparently checks this and does *not* deliver twice to mailboxes. However, devices and servers do *not* have mailboxes to check against. Servers can have some understanding of special mail baskets in the Postmaster's mailbox and can be written to check for duplicate deliveries.



For more information server baskets, please refer to the XMAIC entry points in the "Server Message Activities" chapter in the "Classic MailMan APIs" section of this manual.

Appendix B—Efficient Use of the API

Technical Background

Large messages can be created in a more efficient way using the following method of creating a message.

The simplest and most straightforward approach of creating a message using the API is as follows:

- Load the text of a message into an array
- Set a couple of variables
- D ^XMD

With short messages and if a local array is used, this is fairly efficient. However, when a large message is built, it usually must be stored in a global array. Then, MailMan must read it and rewrite it. This effectively doubles the amount of work the system must do to compile the text of the message. Also, "KILLing" the temporary global array built to store the data passed to the MailMan programmer interface eats up additional resources.

Thus, why not write the text of the message (the data) directly into the message using the available and documented entry points?

Example

The following steps assume that the standard variables already exist in the partition from either Logon or because the job is a TaskMan task.

Step 1—Create a Message with No Text

```
S XMSUB="LARGE DATA TRANSMISSION"; Initialize Subject
S XMDUZ="Application X"           ; Sender
D XMZ^XMA2                       ; Call Create Message Module
I XMZ<1 G RETRY                   ; Abort or retry, if returned value <1
                                   ; Make sure you check!
```

Step 2—Put Text into Message

```
S L=0                             ; Initialize Line Counter
A S L=L+1                          ; Increment Line Counter
D data^routine                     ; Create a Line of Data for the Message in 'X'
I $L(X) S ^XMB(3.9,XMZ,2,L,0)=X    ; Put Line of Data into Message
S ^XMB(3.9,XMZ,2,0)="^3.92^"_L_"^"_L_"^"_DT
```

Step 3—Deliver Message to Recipients

```
S XMDUZ="SENDER,LARGEMESSAGE";   A Sender can be free text or you can
                                ;   Leave the variable undefined and the
                                ;   message will appear to be from the
                                ;   user who was logged on.
S XMY("XXX@Q-AUSTIN_'Q'_DOMAIN")="" ; Remote Recipient
S XMY(234567)=""                 ;   Individual as a recipient
.
.
.
D ENT1^XMD ;                       Call for MailMan Delivery
```

The message will now be delivered. This may not happen immediately because the job of delivery the message is passed off to a 'background filer'.

Appendix C—Looking Up Messages

DIC Lookups

Using DIC lookups into the MESSAGE file (#3.9) is a simple process. The most likely way to do this is by message number (e.g., "123456") or by message subject. In the case where looking the message up by subject is required, the code should understand that message subjects are stored in a case sensitive fashion and the case sensitivity is carried over the network. Putting a "Z" into DIC(0) returns the entire zero node, if it is required. However, the data returned in Y(0) will be raw data and there may be problems with using it directly. It is recommended that calls to the MailMan API be used when extracting data from a field is necessary.

When processing the text of a message, it is recommended that the MailMan Message Server Protocol be used. This means that a message is sent to a serve (S.option_name). When the message is received by the server, which is really a piece of software—an option in the OPTION file (#19), the server protocol can be used. There is also a way to set up and use the executable variable XMREC if needed.



For more information on the MailMan Message Server Protocol, please refer to "Appendix A—Message Server Protocol" in this manual.

Since DIC ensures that only the sender and recipients of a message will be able to find them, the DIC lookup should be used at all times.

Appendix D—Setting Up Bulletins

What is a Bulletin?

A bulletin consists of a form that may be filled in and an optional mail group. The mail group is normally the first thing that is set up. When code is invoked to send a bulletin, the bulletin will be sent to all the members of that Mail Group. Messages (created from bulletins) can be sent very easily and can be triggered by events in the database, or by code written in a routine by a programmer as in the example of a bulletin cross-reference and an API call (see the examples that follow).

First, in order, set up the entries in the MAIL GROUP file (#3.8) and then in the BULLETIN file (#3.6).

The following is an example of a bulletin:

```
NAME:XMTEST      SUBJECT:TEST BULLETIN TYPE |2|
MESSAGE:  Test Bulletin has been triggered by |1|.
MAIL GROUP:POSTMASTER
DESCRIPTION:  THIS IS A TEST BULLETIN
```

Figure 1: An Example of a Bulletin

Calling the Programmer API to Send a Bulletin

Variable input into the message text and subject of the message created with a call to the Bulletin API is arranged by the creation of parameters as one enters data into the bulletin fields. Note that there is only one parameter in the message text portion and the subject portion. That parameter is indicated in the MESSAGE field of the BULLETIN file (#3.6) entry by the string "|1|" and the SUBJECT field with the string "|2|" (see Figure 1). There can be additional parameters. It is only necessary to enter them into the fields. This bulletin could have had much longer text because the MESSAGE field is a WORD-PROCESSING-type field and the variable portion of the text can be on any line of the text, *not* just the first line.

The programmer call to the Bulletin API would then look like this:

```
S XMB="XMTST"
S XMB(1)="JONES, THOMAS"
S XMB(2)="**User Notification**"
S XMDUZ="ApplicationX"
D ^XMB
```



Compare to EN^XMB in Chapter 8 and SENDBULL^XMXAPI and TASKBULL^XMXAPI in Chapter 16 in this manual.

Using the example in Figure 1, this call to the Bulletin API would result in the following:

- A message being sent to the G.POSTMASTER mail group.
- Subject—"TEST BULLETIN TYPE**User Notification**".
- Text—"Test bulletin has been triggered by JONES,THOMAS".



The windows (i.e., |1| and |2| in Figure 1 have been replaced by the values of the corresponding nodes of the XMB Array.

Bulletin Cross-reference

The following dialogue illustrates setting up a Bulletin Cross-reference:

```

Select VA FileMan Option: ENTER or Edit File Entries
INPUT TO WHAT FILE:// BULLETIN
EDIT WHICH FIELD:ALL// <RET>

Select BULLETIN NAME: AFS INSURANCE
Located in the A (Local) namespace.
Are you adding 'AFS INSURANCE' as a new BULLETIN (THE 85TH)? Y <RET> (YES)
SUBJECT: HEALTH INS. FOR PATIENT
Select MAIL GROUP: MCCR DATA COLLECTION
DESCRIPTION:
1>THIS IS A LOCAL BULLETIN. IT NOTIFIES MEMBERS OF THE MCCR DATA
2>COLLECTION MAIL GROUP WHENEVER THE COVERED BY HEALTH INSURANCE
3>QUESTION IN THE PATIENT FILE IS ANSWERED YES.
4><RET>
EDIT Option: <RET>
MESSAGE:
1> PATIENT|1|, SSN#|2|
2>
3>Has had the question, 'COVERED BY HEALTH INSURANCE?' answered 'YES'
4>during the LOAD/EDIT PATIENT DATA process.
5><RET>
EDIT Option: <RET>
Select PARAMETER: 1
DESCRIPTION:
1>PATIENT PATIENT'S NAME from File 2
2><RET>
EDIT Option: <RET>
Select PARAMETER: 2
DESCRIPTION:
1>PATIENT'S SSN from File 2
2><RET>
EDIT Option: <RET>
Select PARAMETER: <RET>

Select BULLETIN NAME: <RET>

Select VA FileMan Option: UTILIY FUNCTIONS

Select Utility Functions Option: CROSS-Reference a File or Field

MODIFY WHAT FILE:BULLETIN// 2 <RET> PATIENT (27866 Entries)
Select FIELD: COVERED BY HEALTH INSURANCE

NO CURRENT CROSS-REFERENCE
WANT TO CREATE A NEW CROSS-REFERENCE FOR THIS FIELD? NO// Y <RET> (YES)
CROSS-REFERENCE NUMBER:1// <RET>
Select TYPE OF INDEXING:REGULAR// BULLETIN

```

Figure 2: An Example of Setting Up a Bulletin Cross-reference

Figure 2 (continued):

```

---SET LOGIC---

ENTER THE NAME OF A 'BULLETIN' MESSAGE, IF YOU WANT THAT MESSAGE SENT WHENEVER
THE 'COVERED BY HEALTH INSURANCE?' FIELD IS ENTERED OR CHANGED:AFS INSURANCE
MESSAGE:
1>  PATIENT |1|, SSN#|2|
2>
3>Has had the question, "COVERED BY HEALTH INSURANCE?" answered 'YES'
4>during the LOAD/EDIT PATIENT DATA process.
5><RET>
EDIT Option: <RET>

DO YOU WANT TO MAKE THE SENDING OF 'AFS INSURANCE CONDITIONAL? NO/ Y <RET> (YES)
ENTER AN EXPRESSION FOR THE CONDITION:COVERED BY HEALTH INSURANCE? ["Y" <RET>
...OK

ENTER A FIELD NAME (FOR EXAMPLE, 'COVERED BY HEALTH INSURANCE?'), OR A COMPUTED-
FIELD EXPRESSION, THE VALUE OF WHICH WILL BE PASSED INTO THE 'AFS INSURANCE'
MESSAGE, AS PARAMETER #2
  --PATIENT's SSN from File 2
PARAMETER #2: SOCIAL SECURITY NUMBER <RET> ...OK

NOW, IF THE BULLETIN IS TO HAVE 3 OR MORE PARAMETERS INSERTED, ENTER A FIELD NAME
(FOR EXAMPLE, 'COVERED BY HEALTH INSURANCE?'), OR A 'COMPUTED-FIELD' EXPRESSION,
THE VALUE OF WHICH WILL BE PASSED INTO THE 'AFS INSURANCE' MESSAGE, AS PARAMETER
#3
(NOTE THAT NO SUCH PARAMETER IS DEFINED FOR THE 'AFS INSURANCE' BULLETIN)
PARAMETER #3:

---KILL LOGIC---

ENTER THE NAME OF A 'BULLETIN' MESSAGE IF YOU WANT THAT MESSAGE SENT WHENEVER THE
'COVERED BY HEALTH INSURANCE?' FIELD IS CHANGED OR DELETED: <RET> NO EFFECT

...CROSS-REFERENCE IS SET

DO YOU WANT TO RUN THE CROSS-REFERENCE FOR EXISTING ENTRIES NOW? NO// N <RET>
(NO)

```

Figure 2: An Example of Setting Up a Bulletin Cross-reference (continued)

Index

A

Address Lookup

- APIs, 9-1
- DES^XMA21, 9-1
- DEST^XMA21, 9-1
- INST^XMA21, 9-2
- WHO^XMA21, 9-3

ADDRNSND^XMXAPI, 16-15

ANSRMSG^XMXAPI, 16-5

APIs

- \$\$ACCESS^XMXSEC, 22-1
- \$\$ANSWER^XMXSEC, 22-1
- \$\$BCAST^XMXSEC, 22-2
- \$\$BMSGCT^XMXUTIL, 23-1
- \$\$BNMSGCT^XMXUTIL, 23-1
- \$\$BSKT^XMAD2, 12-5
- \$\$BSKT^XMXUTIL2, 25-1
- \$\$BSKTNAME^XMXUTIL, 23-1
- \$\$CLOSED^XMXSEC, 22-2
- \$\$CONFID^XMXSEC, 22-2
- \$\$CONFIRM^XMXSEC, 22-2
- \$\$CONVERT^XMXUTIL1, 24-1
- \$\$COPY^XMXSEC, 22-3
- \$\$COPYAMT^XMXSEC1, 22-11
- \$\$COPYLIMS^XMXSEC1, 22-11
- \$\$COPYRECP^XMXSEC1, 22-11
- \$\$CTRL^XMXUTIL1, 24-1
- \$\$DATE^XMXUTIL2, 25-1
- \$\$DECODEUP^XMCU1, 3-2
- \$\$DECODEUP^XMXUTIL1, 24-1
- \$\$DELETE^XMXSEC, 22-3
- \$\$DM^XMBGRP, 14-1
- \$\$EDIT^XMXSEC2, 22-15
- \$\$ENCODEUP^XMCU1, 3-3
- \$\$ENCODEUP^XMXUTIL1, 24-1
- \$\$ENT^XMA2R, 7-1
- \$\$ENTA^XMA2R, 7-2
- \$\$FORWARD^XMXSEC, 22-3
- \$\$FROM^XMXUTIL2, 25-2
- \$\$GMTDIFF^XMXUTIL1, 24-2
- \$\$GOTLOCAL^XMXAPIG, 19-1
- \$\$HDR^XMGAPI2, 10-1
- \$\$INDT^XMXUTIL1, 24-2
- \$\$INFO^XMA11, 4-2
- \$\$INFO^XMXSEC, 22-4
- \$\$KSEQN^XMXUTIL2, 25-2

- \$\$LATER^XMXSEC, 22-4
- \$\$LINE^XMXUTIL2, 25-2
- \$\$MAXBLANK^XMXUTIL1, 24-2
- \$\$MELD^XMXUTIL1, 24-2
- \$\$MG^XMBGRP, 14-2
- \$\$MMDT^XMXUTIL1, 24-3
- \$\$MOVE^XMXSEC, 22-4
- \$\$NAME^XMXUTIL, 23-2
- \$\$NET^XMRENT, 10-3
- \$\$NETNAME^XMXUTIL, 23-2
- \$\$NEW^XMXUTIL2, 25-2
- \$\$NEWS^XMXUTIL, 23-2
- \$\$NU^XM, 12-1
- \$\$ORIGIN8R^XMXSEC, 22-4
- \$\$PAKMAN^XMXSEC1, 22-12
- \$\$POSTPRIV^XMXSEC, 22-5
- \$\$PRI^XMXUTIL2, 25-3
- \$\$PRIORITY^XMXSEC, 22-5
- \$\$QRESP^XMXUTIL2, 25-3
- \$\$READ^XMGAPI1, 11-1
- \$\$READ^XMXSEC, 22-5
- \$\$REN^XMA03, 12-3
- \$\$REPLY^XMXSEC, 22-5
- \$\$RESP^XMXUTIL2, 25-3
- \$\$RPRIV^XMXSEC, 22-6
- \$\$RTRAN^XMCU1, 3-3
- \$\$RWPRIV^XMXSEC, 22-6
- \$\$SCRUB^XMXUTIL1, 24-3
- \$\$SEND^XMXSEC, 22-6
- \$\$SRVTIME^XMS1, 13-2
- \$\$STATUS^XMS1, 13-2
- \$\$STRAN^XMCU1, 3-3
- \$\$STRIP^XMXUTIL1, 24-3
- \$\$SUBCHK^XMGAPI0, 4-5
- \$\$SUBGET^XMGAPI0, 10-1
- \$\$SUBJ^XMXUTIL2, 25-4
- \$\$SURRACC^XMXSEC, 22-6
- \$\$SURRCONF^XMXSEC, 22-7
- \$\$TIMEDIFF^XMXUTIL1, 24-3
- \$\$TMSGCT^XMXUTIL, 23-3
- \$\$TNMSGCT^XMXUTIL, 23-3
- \$\$TSTAMP^XMXUTIL1, 24-3
- \$\$WPRIV^XMXSEC, 22-7
- \$\$ZCLOSED^XMXSEC, 22-7
- \$\$ZCONFID^XMXSEC, 22-8
- \$\$ZCONFIRM^XMXSEC, 22-8
- \$\$ZDATE^XMXUTIL2, 25-4
- \$\$ZFROM^XMXUTIL2, 25-4

- \$\$ZINFO^XMXSEC, 22-8
- \$\$ZNODE^XMXUTIL2, 25-4
- \$\$ZORIGIN8^XMXSEC, 22-9
- \$\$ZPOSTPRV^XMXSEC, 22-9
- \$\$ZPRI^XMXSEC, 22-9
- \$\$ZPRI^XMXUTIL2, 25-5
- \$\$ZREAD^XMXUTIL2, 25-5
- \$\$ZSUBJ^XMXUTIL2, 25-5
- ^XMAH1, 7-3
- ^XMB, 8-1
- ^XMD, 5-1
- Address Lookup, 9-1
- ADDRNSND^XMXAPI, 16-15
- ANSRMSG^XMXAPI, 16-5
- Basket Actions (^XMXAPIB), 18-1
- BULL^XMB, 8-2
- CHECKIN^XM, 20-1
- CHECKOUT^XM, 20-1
- CHK^XMA21, 14-1
- CHKLINES^XMXSEC1, 22-12
- CHKMSG^XMXSEC1, 22-12
- Classic MailMan APIs (Pre-Patch 50), I-1
- CLOSED^XMXEDIT, 21-1
- CONFID^XMXEDIT, 21-2
- CONFIRM^XMXEDIT, 21-2
- CRE8BSKT^XMXAPIB, 18-1
- CRE8MBOX^XMXAPIB, 17-1
- CRE8XMZ^XMXAPI, 16-15
- Creating a Message, 4-1
 - Common Variables, 4-1
- Creating and Sending a Bulletin, 8-1
- Creating and Sending a PackMan Message, 6-1
- Creating and Sending a Reply or Answer, 7-1
- Creating/Sending/Forwarding a Message, 5-1
- Date and String Utility Functions and Routines (^XMXUTIL1), 24-1
- DELBKT^XMXAPIB, 18-2
- DELIVER^XMXEDIT, 21-3
- DELMSG^XMXAPI, 16-6
- DES^XMA21, 9-1
- DEST^XMA21, 9-1
- EN^XM, 20-1
- EN^XMB, 8-2
- EN1^XMD, 5-3
- ENL^XMD, 5-4
- ENT^XMD, 5-4
- ENT^XMPG, 6-1
- ENT^XMUT7, 3-3
- ENT1^XMD, 5-5
- ENT2^XMD, 5-5
- ENT8^XMAH, 10-1
- ENTA^XMAH1, 7-3
- ENTPRT^XMA0, 12-2
- Errors, 16-1, 17-1, 18-1, 19-1, 21-1, 22-1
- FLTRBSKT^XMXAPIB, 18-2
- FLTRMBOX^XMXAPIB, 17-1
- FLTRMSG^XMXAPI, 16-6
- FWDMSG^XMXAPI, 16-7
- GET^XMA2, 4-4
- GET^XML, 11-1
- GETRESTR^XMXSEC1, 22-13
- Getting Information About a Message, 10-1
- Getting Text From a Message, 11-1
- GO^XMCTLK, 2-1
- HDR^XMA0, 12-2
- HEADER^XM, 20-1
- INFO^XMXEDIT, 21-3
- INIT^XMVVITAE, 15-1
- INMSG^XMXUTIL2, 25-6
- INMSG1^XMXUTIL2, 25-10
- INMSG2^XMXUTIL2, 25-11
- INRESP^XMXUTIL2, 25-12
- INRESPS^XMXUTIL2, 25-13
- INST^XMA21, 9-2
- Interactive User Actions (^XM & ^XMXAPIU), 20-1
- KILL^XM, 3-1
- KL^XMA1B, 12-4
- KLQ^XMA1B, 12-4
- KVAPOR^XMXUTIL, 23-3
- LASTACC^XMXUTIL, 23-4
- LATERMSG^XMXAPI, 16-7
- LISTBSKT^XMXAPIB, 18-3
- LISTMSG^XMXAPIB, 18-5
- Mail Group Activities, 14-1
- Mail Group Functions and Actions (^XMXAPIG), 19-1
- Mailbox Actions (^XMXAPIB), 17-1
- Mailbox, Basket, and Message Activities, 12-1
- MAKENEW^XMXUTIL, 23-5
- Message Action Building Blocks (^XMXAPI), 16-15
- Message Actions (^XMXAPI), 16-1
- Message and Mailbox Utility Functions and Routines (^XMXUTIL), 23-1
- Message Editing (^XMXEDIT), 21-1
- Message Information Functions and Routines (^XMXUTIL2 & ^XMXUTIL3), 25-1, 25-15

- Message Security, Permissions, and Restrictions (Functions and Routines)—
^XMXSEC, 22-1
 - Message Security, Permissions, and Restrictions (Functions and Routines)—
^XMXSEC1, 22-11
 - Message Security, Permissions, and Restrictions (Functions and Routines)—
^XMXSEC2, 22-15
 - MOVEMSG^XMXAPI, 16-8
 - N1^XM, 3-1
 - NAMEBSKT^XMXAPIB, 18-9
 - New MailMan APIs (Post-Patch 50), II-1
 - NEW^XM, 3-2
 - NONEW^XMXUTIL, 23-5
 - OPTEDIT^XMXSEC2, 22-15
 - OPTGRP^XMXSEC1, 22-14
 - OPTMSG^XMXSEC2, 22-16
 - OTHER^XMOVVITAE, 15-3
 - PAGE^XMXUTIL, 23-6
 - Parameter Definitions, 16-1
 - PR2^XMA0, 12-3
 - PRIORITY^XMXEDIT, 21-4
 - Programmer APIs, 1
 - Programmer Calls, 2-1
 - PRTMSG^XMXAPI, 16-9
 - PUTSERV^XMXAPI, 16-10
 - Q^XMXUTIL3, 25-15
 - QBSKT^XMXAPIB, 18-9
 - QD^XMXUTIL3, 25-16
 - QL^XMXUTIL3, 25-19
 - QMBOX^XMXAPIB, 17-2
 - QN^XMXUTIL3, 25-20
 - READ^XMXAPIU, 20-3
 - READNEW^XMXAPIU, 20-3
 - REC^XMA, 12-1
 - REC^XMS3, 11-2
 - REMSBMSG^XMA1C, 13-1
 - REPLYMSG^XMXAPI, 16-10
 - RSEQBSKT^XMXAPIB, 18-10
 - S2^XMA1B, 12-5
 - SELF^XMOVVITAE, 15-5
 - SEND^XMXAPIU, 20-4
 - SENBULL^XMXAPI, 16-11
 - SENDMSG^XMXAPI, 16-12
 - Server Message Activities, 13-1
 - SETSB^XMA1C, 13-1
 - SUBJ^XMXEDIT, 21-4
 - TASKBULL^XMXAPI, 16-13
 - TERMMBOX^XMXAPIB, 17-3
 - TERMMSG^XMXAPI, 16-14
 - TEXT^XMXEDIT, 21-5
 - TOWHOM^XMXAPI, 16-16
 - TOWHOM^XMXAPIU, 20-4
 - User Information (^XMOVVITAE), 15-1
 - Utility Calls, 3-1
 - VAPOR^XMXEDIT, 21-5
 - VSUBJ^XMXAPI, 16-16
 - WAIT^XMXUTIL, 23-6
 - WHO^XMA21, 9-3
 - WRITE^XMA11A, 5-1, 7-1
 - XM, 2-1
 - XMZ^XMA2, 4-3
 - ZAPSERV^XMXAPI, 16-14
 - ZONEDIFF^XMXUTIL1, 24-4
 - ZTSK^XMADGO, 3-2
 - Assumptions About the Reader, 2
- B**
- Basket Actions (^XMXAPIB), 18-1
 - CRE8BSKT^XMXAPIB, 18-1
 - DELBSKT^XMXAPIB, 18-2
 - FLTRBSKT^XMXAPIB, 18-2
 - LISTBSKT^XMXAPIB, 18-3
 - LISTMSG^XMXAPIB, 18-5
 - NAMEBSKT^XMXAPIB, 18-9
 - QBSKT^XMXAPIB, 18-9
 - RSEQBSKT^XMXAPIB, 18-10
 - Baskets
 - APIs, 18-1
 - BULL^XMB, 8-2
 - BULLETIN file, 8-1, 8-2, 8-3, 1
 - Bulletins
 - APIs, 8-1
- C**
- CHECKIN^XM, 20-1
 - CHECKOUT^XM, 20-1
 - CHK^XMA21, 14-1
 - CHKLINES^XMXSEC1, 22-12
 - CHKMSG^XMXSEC1, 22-12
 - Classic MailMan APIs
 - Retired, 1-1
 - Classic MailMan APIs (Pre-Patch 50), I-1
 - CLOSED^XMXEDIT, 21-1
 - COMMUNICATIONS PROTOCOL file, 11-1
 - CONFID^XMXEDIT, 21-2
 - CONFIRM^XMXEDIT, 21-2
 - CRE8BSKT^XMXAPIB, 18-1
 - CRE8MBOX^XMXAPIB, 17-1

CRE8XMZ^XMXAPI, 16-15

Creating a Message

\$\$INFO^XMA11, 4-2

\$\$SUBCHK^XMGAPI0, 4-5

APIs, 4-1

Common Variables, 4-1

GET^XMA2, 4-4

XMZ^XMA2, 4-3

Creating and Sending a Bulletin

^XMB, 8-1

APIs, 8-1

BULL^XMB, 8-2

EN^XMB, 8-2

Creating and Sending a PackMan Message

APIs, 6-1

ENT^XMPG, 6-1

Creating and Sending a Reply or Answer

\$\$ENT^XMA2R, 7-1

\$\$ENTA^XMA2R, 7-2

^XMAH1, 7-3

APIs, 7-1

ENTA^XMAH1, 7-3

WRITE^XMA11A, 7-1

Creating/Sending/Forwarding a Message

^XMD, 5-1

APIs, 5-1

EN1^XMD, 5-3

ENL^XMD, 5-4

ENT^XMD, 5-4

ENT1^XMD, 5-5

ENT2^XMD, 5-5

WRITE^XMA11A, 5-1

D

Date and String Utility Functions and Routines

(^XMXUTIL1), 24-1

\$\$CONVERT^XMXUTIL1, 24-1

\$\$CTRL^XMXUTIL1, 24-1

\$\$DECODEUP^XMXUTIL1, 24-1

\$\$ENCODEUP^XMXUTIL1, 24-1

\$\$GMTDIFF^XMXUTIL1, 24-2

\$\$INDT^XMXUTIL1, 24-2

\$\$MAXBLANK^XMXUTIL1, 24-2

\$\$MELD^XMXUTIL1, 24-2

\$\$MMDT^XMXUTIL1, 24-3

\$\$SCRUB^XMXUTIL1, 24-3

\$\$STRIP^XMXUTIL1, 24-3

\$\$TIMEDIFF^XMXUTIL1, 24-3

\$\$TSTAMP^XMXUTIL1, 24-3

ZONEDIFF^XMXUTIL1, 24-4

DELBSKT^XMXAPIB, 18-2

DELIVER^XMXEDIT, 21-3

DELMSG^XMXAPI, 16-6

DES^XMA21, 9-1

DEST^XMA21, 9-1

Developer APIs, 1

Documentation Symbols, 1

DOMAIN file, 25-18

E

EN^XM, 20-1

EN^XMB, 8-2

EN1^XMD, 5-3

ENL^XMD, 5-4

ENT^XMD, 5-4

ENT^XMPG, 6-1

ENT^XMUT7, 3-3

ENT1^XMD, 5-5

ENT2^XMD, 5-5

ENT8^XMAH, 10-1

ENTA^XMAH1, 7-3

ENTPRT^XMA0, 12-2

Errors, 16-1, 17-1, 18-1, 19-1, 21-1, 22-1

F

FIND^DIC, 25-15, 25-16

FLTRBSKT^XMXAPIB, 18-2

FLTRMBOX^XMXAPIB, 17-1

FLTRMSG^XMXAPI, 16-6

FWDMSG^XMXAPI, 16-7

G

GET^XMA2, 4-4

GET^XML, 11-1

GETRESTR^XMXSEC1, 22-13

Getting Information About a Message

\$\$HDR^XMGAPI2, 10-1

\$\$NET^XMRENT, 10-3

\$\$SUBGET^XMGAPI0, 10-1

APIs, 10-1

ENT8^XMAH, 10-1

Getting Text From a Message

\$\$READ^XMGAPI1, 11-1

APIs, 11-1

GET^XML, 11-1

REC^XMS3, 11-2

Glossary, 1

GO^XMCTLK, 2-1

H

HDR^XMA0, 12-2
 HEADER^XM, 20-1
 Home Page, MailMan Home Page Web Address,
 2
 How To
 Use this Manual, 1

I

INFO^XMXEDIT, 21-3
 INIT^XMVITAE, 20-3
 INIT^XMVVITAE, 15-1, 21-1
 INMSG^XMXUTIL2, 25-6, 25-8
 INMSG1^XMXUTIL2, 22-16, 23-4, 25-6, 25-7,
 25-10, 25-12, 25-13
 INMSG2^XMXUTIL2, 21-1, 22-16, 23-4, 25-6,
 25-7, 25-11
 INRESP^XMXUTIL2, 25-12
 INRESPTS^XMXUTIL2, 23-4, 25-10, 25-13
 INST^XMA21, 9-2
 Interactive User Actions (^XM & ^XMXAPIU),
 20-1
 CHECKIN^XM, 20-1
 CHECKOUT^XM, 20-1
 EN^XM, 20-1
 HEADER^XM, 20-1
 READ^XMXAPIU, 20-3
 READNEW^XMXAPIU, 20-3
 SEND^XMXAPIU, 20-4
 TOWHOM^XMXAPIU, 20-4
 Introduction to MailMan Programmer Manual, 1

K

KILL^XM, 3-1
 KL^XMA1B, 12-4
 KLQ^XMA1B, 12-4
 KVAPOR^XMXUTIL, 21-5, 23-3

L

LAST NEW MSG NOTIFY DATE/TIME field,
 23-3
 LASTACC^XMXUTIL, 23-4
 LATERMSG^XMXAPI, 16-7
 LIST^DIC, 18-5, 25-15, 25-16, 25-19, 25-20
 LISTBSKT^XMXAPIB, 18-3
 Lister, 25-15, 25-19
 LISTMSGS^XMXAPIB, 18-5

M

Mail Group Activities
 \$\$DM^XMBGRP, 14-1
 \$\$MG^XMBGRP, 14-2
 APIs, 14-1
 CHK^XMA21, 14-1
 MAIL GROUP file, 14-1, 1
 Mail Group Functions and Actions
 (^XMXAPIG), 19-1
 \$\$GOTLOCAL^XMXAPIG, 19-1
 Mailbox
 Actions (^XMXAPIB), 17-1
 CRE8MBOX^XMXAPIB, 17-1
 FLTRMBOX^XMXAPIB, 17-1
 QMBOX^XMXAPIB, 17-2
 TERMMBOX^XMXAPIB, 17-3
 MAILBOX File, 23-3
 Mailbox, Basket, and Message Activities
 \$\$BSKT^XMAD2, 12-5
 \$\$NU^XM, 12-1
 \$\$REN^XMA03, 12-3
 APIs, 12-1
 ENTPRT^XMA0, 12-2
 HDR^XMA0, 12-2
 KL^XMA1B, 12-4
 KLQ^XMA1B, 12-4
 PR2^XMA0, 12-3
 REC^XMA, 12-1
 S2^XMA1B, 12-5
 MailMan Home Page Web Address, 2
 MAKENEW^XMXUTIL, 23-5
 Manuals (related), 2
 Message Actions
 Building Blocks (^XMXAPI), 16-15
 ADDRNSND^XMXAPI, 16-15
 CRE8XMZ^XMXAPI, 16-15
 TOWHOM^XMXAPI, 16-16
 VSUBJ^XMXAPI, 16-16
 Message Actions (^XMXAPI), 16-1
 ANSRMSG^XMXAPI, 16-5
 DELMSG^XMXAPI, 16-6
 FLTRMSG^XMXAPI, 16-6
 FWDMSG^XMXAPI, 16-7
 LATERMSG^XMXAPI, 16-7
 MOVEMSG^XMXAPI, 16-8
 PRTMSG^XMXAPI, 16-9
 PUTSERV^XMXAPI, 16-10
 REPLYMSG^XMXAPI, 16-10
 SENDBULL^XMXAPI, 16-11
 SENDMSG^XMXAPI, 16-12

- TASKBULL^XMXAPI, 16-13
- TERMMSG^XMXAPI, 16-14
- ZAPSERV^XMXAPI, 16-14
- Message and Mailbox Utility Functions and Routines (^XMXUTIL), 23-1
 - \$\$BMSGCT^XMXUTIL, 23-1
 - \$\$BNMSGCT^XMXUTIL, 23-1
 - \$\$BSKTNAME^XMXUTIL, 23-1
 - \$\$NAME^XMXUTIL, 23-2
 - \$\$NETNAME^XMXUTIL, 23-2
 - \$\$NEWS^XMXUTIL, 23-2
 - \$\$TMSGCT^XMXUTIL, 23-3
 - \$\$TNMSGCT^XMXUTIL, 23-3
 - KVAPOR^XMXUTIL, 23-3
 - LASTACC^XMXUTIL, 23-4
 - MAKENEW^XMXUTIL, 23-5
 - NONEW^XMXUTIL, 23-5
 - PAGE^XMXUTIL, 23-6
 - WAIT^XMXUTIL, 23-6
- Message Editing (^XMXEDIT), 21-1
 - CLOSED^XMXEDIT, 21-1
 - CONFID^XMXEDIT, 21-2
 - CONFIRM^XMXEDIT, 21-2
 - DELIVER^XMXEDIT, 21-3
 - INFO^XMXEDIT, 21-3
 - PRIORITY^XMXEDIT, 21-4
 - SUBJ^XMXEDIT, 21-4
 - TEXT^XMXEDIT, 21-5
 - VAPOR^XMXEDIT, 21-5
- MESSAGE file, 3-1, 3-2, 4-2, 4-3, 4-4, 5-2, 5-3, 5-4, 5-5, 5-6, 6-1, 7-1, 7-2, 7-3, 8-3, 10-1, 10-2, 10-3, 11-2, 12-2, 12-3, 12-4, 12-5, 13-1, 13-2, 16-4, 16-5, 16-6, 16-10, 16-11, 16-12, 16-14, 16-15, 16-16, 20-4, 21-1, 21-2, 21-3, 21-4, 21-5, 22-1, 22-2, 22-3, 22-4, 22-5, 22-7, 22-8, 22-9, 22-11, 22-12, 22-13, 22-15, 22-16, 23-3, 23-4, 23-5, 25-1, 25-2, 25-3, 25-4, 25-5, 25-6, 25-10, 25-11, 25-12, 25-13, 25-15, 25-16, 25-19, 25-20
- MESSAGE File, 16-1, 16-2, 16-15, 18-5
- Message Information Functions and Routines (^XMXUTIL2 & ^XMXUTIL3), 25-1, 25-15
 - \$\$BSKT^XMXUTIL2, 25-1
 - \$\$DATE^XMXUTIL2, 25-1
 - \$\$FROM^XMXUTIL2, 25-2
 - \$\$KSEQN^XMXUTIL2, 25-2
 - \$\$LINE^XMXUTIL2, 25-2
 - \$\$NEW^XMXUTIL2, 25-2
 - \$\$PRI^XMXUTIL2, 25-3
 - \$\$QRESP^XMXUTIL2, 25-3
 - \$\$RESP^XMXUTIL2, 25-3
 - \$\$SUBJ^XMXUTIL2, 25-4
 - \$\$ZDATE^XMXUTIL2, 25-4
 - \$\$ZFROM^XMXUTIL2, 25-4
 - \$\$ZNODE^XMXUTIL2, 25-4
 - \$\$ZPRI^XMXUTIL2, 25-5
 - \$\$ZREAD^XMXUTIL2, 25-5
 - \$\$ZSUBJ^XMXUTIL2, 25-5
 - INMSG^XMXUTIL2, 25-6
 - INMSG1^XMXUTIL2, 25-10
 - INMSG2^XMXUTIL2, 25-11
 - INRESP^XMXUTIL2, 25-12
 - INRESPS^XMXUTIL2, 25-13
 - Q^XMXUTIL3, 25-15
 - QD^XMXUTIL3, 25-16
 - QL^XMXUTIL3, 25-19
 - QN^XMXUTIL3, 25-20
- Message Security, Permissions, and Restrictions (Functions and Routines)—^XMXSEC, 22-1
 - \$\$ACCESS^XMXSEC, 22-1
 - \$\$ANSWER^XMXSEC, 22-1
 - \$\$BCAST^XMXSEC, 22-2
 - \$\$CLOSED^XMXSEC, 22-2
 - \$\$CONFID^XMXSEC, 22-2
 - \$\$CONFIRM^XMXSEC, 22-2
 - \$\$COPY^XMXSEC, 22-3
 - \$\$DELETE^XMXSEC, 22-3
 - \$\$FORWARD^XMXSEC, 22-3
 - \$\$INFO^XMXSEC, 22-4
 - \$\$LATER^XMXSEC, 22-4
 - \$\$MOVE^XMXSEC, 22-4
 - \$\$ORIGIN8R^XMXSEC, 22-4
 - \$\$POSTPRIV^XMXSEC, 22-5
 - \$\$PRIORITY^XMXSEC, 22-5
 - \$\$READ^XMXSEC, 22-5
 - \$\$REPLY^XMXSEC, 22-5
 - \$\$RPRIV^XMXSEC, 22-6
 - \$\$RWPRIV^XMXSEC, 22-6
 - \$\$SEND^XMXSEC, 22-6
 - \$\$SURRACC^XMXSEC, 22-6
 - \$\$SURRCONF^XMXSEC, 22-7
 - \$\$WPRIV^XMXSEC, 22-7
 - \$\$ZCLOSED^XMXSEC, 22-7
 - \$\$ZCONFID^XMXSEC, 22-8
 - \$\$ZCONFIRM^XMXSEC, 22-8
 - \$\$ZINFO^XMXSEC, 22-8
 - \$\$ZORIGIN8^XMXSEC, 22-9
 - \$\$ZPOSTPRV^XMXSEC, 22-9
 - \$\$ZPRI^XMXSEC, 22-9
- Message Security, Permissions, and Restrictions (Functions and Routines)—^XMXSEC1, 22-11

\$\$COPYAMT^XMXSEC1, 22-11
 \$\$COPYLIMS^XMXSEC1, 22-11
 \$\$COPYRECP^XMXSEC1, 22-11
 \$\$PAKMAN^XMXSEC1, 22-12
 CHKLINES^XMXSEC1, 22-12
 CHKMSG^XMXSEC1, 22-12
 GETRESTR^XMXSEC1, 22-13
 OPTGRP^XMXSEC1, 22-14
 Message Security, Permissions, and Restrictions
 (Functions and Routines)—^XMXSEC2, 22-
 15
 \$\$EDIT^XMXSEC2, 22-15
 OPTEDIT^XMXSEC2, 22-15
 OPTMSG^XMXSEC2, 22-16
 MOVEMSG^XMXAPI, 16-8

N

N1^XM, 3-1
 NAMEBSKT^XMXAPIB, 18-9
 Network Signature, 16-5
 New MailMan APIs (Post-Patch 50), II-1
 NEW PERSON file, 23-2
 NEW^XM, 3-2
 NONEW^XMXUTIL, 23-5

O

OBJECT file, 10-3
 OPTEDIT^XMXSEC2, 21-1, 22-15
 OPTGRP^XMXSEC1, 22-14
 Options
 XMPOST, 8-2
 XMREAD, 12-1
 XMSEND, 5-1, 5-4
 OPTMSG^XMXSEC2, 21-1, 22-15, 22-16
 Orientation For MailMan Programmer Manual,
 1
 OTHER^XMOVVITAE, 15-3

P

PackMan
 APIs, 6-1
 PAGE^XMXUTIL, 23-6
 Parameter Definitions, 16-1
 Post-Patch 50
 New MailMan APIs, II-1
 PR2^XMA0, 12-3
 Preface to the MailMan Programmer Manual, iii
 Pre-Patch 50
 Classic MailMan APIs, I-1

PRIORITY^XMXEDIT, 21-4
 Programmer APIs, 1
 Programmer Calls
 APIs, 2-1
 GO^XMXCTLK, 2-1
 XM, 2-1
 PRTMSG^XMXAPI, 16-9
 PUTSERV^XMXAPI, 16-10

Q

Q^XMXUTIL3, 25-15
 QBSKT^XMXAPIB, 18-9
 QD^XMXUTIL3, 25-16
 QL^XMXUTIL3, 25-19
 QMBOX^XMXAPI, 23-2
 QMBOX^XMXAPIB, 17-2
 QN^XMXUTIL3, 25-20

R

READ^XMXAPIU, 20-3
 Reader, Assumptions About the, 2
 READNEW^XMXAPIU, 20-3
 REC^XMA, 12-1
 REC^XMS3, 11-2
 REMSBMSG^XMA1C, 13-1
 REPLYMSG^XMXAPI, 16-10
 Retired APIs, 1-1
 ROUTINE field, 20-3
 RSEQBSKT^XMXAPIB, 18-10

S

S2^XMA1B, 12-5
 Security Keys
 XM GROUP PRIORITY, 22-13
 XMMGR, 17-3, 18-1, 18-2, 18-9, 18-10, 22-
 12
 SELF^XMOVVITAE, 15-5
 SEND^XMXAPIU, 20-4
 SENDBULL^XMXAPI, 16-11
 SENDMSG^XMXAPI, 16-12
 Server Message Activities
 \$\$SRVTIME^XMS1, 13-2
 \$\$STATUS^XMS1, 13-2
 APIs, 13-1
 REMSBMSG^XMA1C, 13-1
 SETSB^XMA1C, 13-1
 SETSB^XMA1C, 13-1
 SHARED,MAIL, 16-3, 16-5, 18-1, 18-2, 18-9,
 18-10, 22-5, 22-9, 22-13

SUBJ^XMXEDIT, 21-4
 Symbols Found in the Documentation, 1

T

Table of Contents For the MailMan Programmer Manual, v
 TASKBULL^XMXAPI, 16-13
 TERMMBOX^XMXAPIB, 17-3
 TERMMSG^XMXAPI, 16-14
 TEXT^XMXEDIT, 21-5
 ^TMP("XMERR", \$J), 16-5, 16-6, 16-7, 16-8, 16-9, 16-10, 16-11, 16-12, 16-13, 16-14, 16-15, 16-16, 17-1, 17-2, 17-3, 18-1, 18-2, 18-3, 18-5, 18-9, 18-10, 19-1, 21-1, 21-2, 21-3, 21-4, 21-5, 22-1, 22-2, 22-3, 22-4, 22-5, 22-6, 22-7, 22-8, 22-9, 22-11, 22-12, 22-13, 22-14, 22-15, 22-16
 ^TMP("XMLIST", \$J), 25-15, 25-16, 25-19, 25-20
 TOWHOM^XMXAPI, 16-16
 TOWHOM^XMXAPIU, 20-4

U

Use this Manual, How to, 1
 User Information (^XMOVVITAE), 15-1
 INIT^XMOVVITAE, 15-1
 OTHER^XMOVVITAE, 15-3
 SELF^XMOVVITAE, 15-5
 Utility Calls
 \$\$DECODEUP^XMCU1, 3-2
 \$\$ENCODEUP^XMCU1, 3-3
 \$\$RTRAN^XMCU1, 3-3
 \$\$STRAN^XMCU1, 3-3
 APIs, 3-1
 ENT^XMUT7, 3-3
 KILL^XM, 3-1
 N1^XM, 3-1
 NEW^XM, 3-2
 ZTSK^XMADGO, 3-2

V

VAPOR^XMXEDIT, 21-5
 VSUBJ^XMXAPI, 16-16

W

WAIT^XMXUTIL, 23-6
 Web Page, MailMan Home Page Web Address, 2

WHO^XMA21, 9-3
 WRITE^XMA11A, 5-1, 7-1

X

XM, 2-1, 3-1, 3-2, 12-1, 20-1
 XM GROUP PRIORITY Security Key, 22-13
 XMA, 12-1, 12-2
 XMA0, 12-2, 12-3
 XMA03, 12-3
 XMA11, 4-2
 XMA11A, 5-1, 7-1
 XMA1B, 12-4, 12-5
 XMA1C, 13-1
 XMA2, 4-3, 4-4
 XMA21, 9-1, 9-2, 9-3, 14-1
 XMA2R, 7-1, 7-2
 XMAD2, 12-5
 XMADGO, 3-2
 XMAH, 10-1
 XMAH1, 7-3
 XMB, 8-1, 8-2
 XMBGRP, 14-1, 14-2
 XMCTLK, 2-1
 XMCU1, 3-2, 3-3
 XMD, 5-1, 5-3, 5-4, 5-5
 XMERR, 16-5, 16-6, 16-7, 16-8, 16-9, 16-10, 16-11, 16-12, 16-13, 16-14, 16-15, 16-16, 17-1, 17-2, 17-3, 18-1, 18-2, 18-3, 18-5, 18-9, 18-10, 19-1, 21-1, 21-2, 21-3, 21-4, 21-5, 22-1, 22-2, 22-3, 22-4, 22-5, 22-6, 22-7, 22-8, 22-9, 22-11, 22-12, 22-13, 22-14, 22-15, 22-16
 XMGAPI0, 4-5, 10-1
 XMGAPI1, 11-1
 XMGAPI2, 10-1
 XMINSTR, 21-1
 XML, 11-1
 XMMGR Security Key, 17-3, 18-1, 18-2, 18-9, 18-10, 22-12
 XMPG, 6-1
 XMPOST option, 8-2
 XMREAD option, 12-1
 XMRENT, 10-3
 XMS1, 13-2
 XMS3, 11-2
 XMSEND option, 5-1, 5-4
 XMUT7, 3-3
 XMV array, 21-1
 XMOVVITAE, 15-1, 15-3, 15-5

XMXAPI, 16-1, 16-5, 16-6, 16-7, 16-8, 16-9,
 16-10, 16-11, 16-12, 16-13, 16-14, 16-15, 16-
 16
 XMXAPIB, 17-1, 17-2, 17-3, 18-1, 18-2, 18-3,
 18-5, 18-9, 18-10
 XMXAPIG, 19-1
 XMXAPIU, 20-1, 20-3, 20-4
 XMXEDIT, 21-1, 21-2, 21-3, 21-4, 21-5
 XMXSEC, 22-1, 22-2, 22-3, 22-4, 22-5, 22-6,
 22-7, 22-8, 22-9
 XMXSEC1, 22-11, 22-12, 22-13, 22-14
 XMXSEC2, 22-15, 22-16
 XMXUTIL, 23-1, 23-2, 23-3, 23-4, 23-5, 23-6
 XMXUTIL1, 24-1, 24-2, 24-3, 24-4
 XMXUTIL2, 25-1, 25-2, 25-3, 25-4, 25-5, 25-6,
 25-10, 25-11, 25-12, 25-13
 XMXUTIL3, 25-1, 25-15, 25-16, 25-19, 25-20
 XMZ^XMA2, 4-3

Z

ZAPSERV^XMXAPI, 16-14
 ZONEDIFF^XMXUTIL1, 24-4
 ZTSK^XMADGO, 3-2

\$

\$\$ACCESS^XMXSEC, 21-1, 22-1
 \$\$ANSWER^XMXSEC, 22-1
 \$\$BCAST^XMXSEC, 22-2
 \$\$BMSGCT^XMXUTIL, 23-1
 \$\$BNMSGCT^XMXUTIL, 23-1
 \$\$BSKT^XMAD2, 12-5
 \$\$BSKT^XMXUTIL2, 25-1
 \$\$BSKTNAME^XMXUTIL, 23-1
 \$\$CLOSED^XMXSEC, 22-2
 \$\$CONFID^XMXSEC, 22-2
 \$\$CONFIRM^XMXSEC, 22-2
 \$\$CONVERT^XMXUTIL1, 24-1
 \$\$COPY^XMXSEC, 22-3
 \$\$COPYAMT^XMXSEC1, 22-11
 \$\$COPYLIMS^XMXSEC1, 22-11
 \$\$COPYRECP^XMXSEC1, 22-11
 \$\$CTRL^XMXUTIL1, 24-1
 \$\$DATE^XMXUTIL2, 25-1
 \$\$DECODEUP^XMCU1, 3-2
 \$\$DECODEUP^XMXUTIL1, 24-1
 \$\$DELETE^XMXSEC, 22-3
 \$\$DM^XMBGRP, 14-1
 \$\$EDIT^XMXSEC2, 21-1, 22-15
 \$\$ENCODEUP^XMCU1, 3-3

\$\$ENCODEUP^XMXUTIL1, 24-1
 \$\$ENT^XMA2R, 7-1
 \$\$ENTA^XMA2R, 7-2
 \$\$FORWARD^XMXSEC, 22-3
 \$\$FROM^XMXUTIL2, 25-2
 \$\$GET1^DIQ, 11-1
 \$\$GMTDIFF^XMXUTIL1, 24-2
 \$\$GOTLOCAL^XMXAPIG, 19-1
 \$\$HDR^XMGAPI2, 10-1, 25-8
 \$\$INDT^XMXUTIL1, 24-2
 \$\$INFO^XMA11, 4-2
 \$\$INFO^XMXSEC, 22-4
 \$\$KSEQN^XMXUTIL2, 25-2
 \$\$LATER^XMXSEC, 22-4
 \$\$LINE^XMXUTIL2, 25-2
 \$\$MAXBLANK^XMXUTIL1, 24-2
 \$\$MELD^XMXUTIL1, 24-2
 \$\$MG^XMBGRP, 14-2
 \$\$MMDT^XMXUTIL1, 24-3
 \$\$MOVE^XMXSEC, 22-4
 \$\$NAME^XMXUTIL, 23-2
 \$\$NET^XMRENT, 10-3
 \$\$NETNAME^XMXUTIL, 23-2
 \$\$NEW^XMXUTIL2, 25-2
 \$\$NEWS^XMXUTIL, 23-2
 \$\$NU^XM, 12-1
 \$\$ORIGIN8R^XMXSEC, 22-4, 25-12
 \$\$PAKMAN^XMXSEC1, 22-12
 \$\$POSTPRIV^XMXSEC, 22-5
 \$\$PRI^XMXUTIL2, 25-3
 \$\$PRIORITY^XMXSEC, 22-5
 \$\$QRESP^XMXUTIL2, 25-3
 \$\$READ^XMGAPI1, 11-1
 \$\$READ^XMXSEC, 22-5
 \$\$REN^XMA03, 12-3
 \$\$REPLY^XMXSEC, 22-5
 \$\$RESP^XMXUTIL2, 25-3
 \$\$RPRIV^XMXSEC, 22-6
 \$\$RTRAN^XMCU1, 3-3
 \$\$RWPRIV^XMXSEC, 22-6
 \$\$SCRUB^XMXUTIL1, 24-3
 \$\$SEND^XMXSEC, 22-6
 \$\$SRVTIME^XMS1, 13-2
 \$\$STATUS^XMS1, 13-2
 \$\$STRAN^XMCU1, 3-3
 \$\$STRIP^XMXUTIL1, 24-3
 \$\$SUBCHK^XMGAPI0, 4-5
 \$\$SUBGET^XMGAPI0, 10-1
 \$\$SUBJ^XMXUTIL2, 25-4
 \$\$SURRACC^XMXSEC, 22-6
 \$\$SURRCONF^XMXSEC, 22-7

Index

\$\$TIMEDIFF^XMXUTIL1, 24-3
\$\$TMSGCT^XMXUTIL, 23-3
\$\$TNMSGCT^XMXUTIL, 23-3
\$\$TSTAMP^XMXUTIL1, 24-3
\$\$WPRIV^XMXSEC, 22-7
\$\$ZCLOSED^XMXSEC, 22-7
\$\$ZCONFID^XMXSEC, 22-8
\$\$ZCONFIRM^XMXSEC, 22-8
\$\$ZDATE^XMXUTIL2, 25-4

\$\$ZFROM^XMXUTIL2, 25-4
\$\$ZINFO^XMXSEC, 22-8
\$\$ZNODE^XMXUTIL2, 25-4
\$\$ZORIGIN8^XMXSEC, 22-9
\$\$ZPOSTPRV^XMXSEC, 22-9
\$\$ZPRI^XMXSEC, 22-9
\$\$ZPRI^XMXUTIL2, 25-5
\$\$ZREAD^XMXUTIL2, 25-5
\$\$ZSUBJ^XMXUTIL2, 25-5