MAILMAN

RELEASE NOTES

Version 8.0

August 2002

Revised September 2006

Revision History

Documentation Revisions

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

Date	Revision	Description	Author
07/23/02	1.0	Initial MailMan V. 8.0 software and documentation release. MailMan V. 8.0 was first released as "DNS-Aware MailMan" in a supplemental document released in August 2002. However, the remaining MailMan documentation set was never updated.	Thom Blom and Gary Beuschel Oakland, CA Office of Information Field Office (OIFO)
09/21/06	2.0	MailMan V. 8.0 documentation reformatting/revision. Reformatted document to follow the latest ISS styles and guidelines. As of this date, all content updates have been completed for all released MailMan patches. **PDF 508 Compliance—**The final PDF document supports the minimum requirements to be 508 compliant (i.e., accessibility tags, language selection, alternate text for all images/icons, fully functional Web links, successfully passed Adobe Acrobat Quick Check).	MailMan Development Team Oakland, CA Office of Information Field Office (OIFO): • Maintenance Project Manager—Jack Schram • Project Planner—Laura Rowland • Developers—Gary Beuschel • Technical Writer—Thom Blom

Table i: Documentation revision history

Patch Revisions

For a complete list of patches released with this software, please refer to the Patch Module on FORUM.

Contents

Contents

MailMan Getting Started Guide
Version 8.0 Revised September 2006

MailMan Version 8.0 Release Notes

This document highlights the various new maintenance requirements, technical solutions, and the functionality improvements introduced with MailMan V. 8.0.

The topics covered in this document include:

- Why was MailMan V. 8.0 Created?

- What are the New Features and Improvements?

Why was MailMan V. 8.0 Created?

MailMan, part of the Veterans Health Information Systems and Technology Architecture (VistA), is over 20 years old now. It was one of the first VistA software applications implemented at field facilities. The code was compactly written, and variable names were kept to very few characters. However, over the years, as it grew and gained capabilities, MailMan has been continually patched, making it very difficult to maintain and was lacking certain functionality. Thus, a new version of MailMan was created in order to ease installation and maintenance and to introduce new functionality.

What are the New Features and Improvements?

MailMan V. 8.0 is DNS-Aware

MailMan V. 8.0 is now DNS-Aware. It uses Kernel's MAIL^XLFNSLK API to retrieve IP addresses. Thus, it is no longer necessary to manually update the IP addresses in the DOMAIN file (#4.2). The IP address fields still remain in File #4.2 and MailMan continues to use them. However, if they don't work, MailMan uses the Kernel API to retrieve a list of valid IP addresses. When MailMan finds one that works, MailMan replaces the non-working IP address with the working one.

In order to activate DNS awareness, the new DNS AWARE field (#8.22) in the MAILMAN SITE PARAMETERS file (#4.3) *must* be set to Yes.

Also, routine ^XLFNSLK *must* exist, and the DNS IP field (#51) in the KERNEL SYSTEM PARAMETERS file (#8989.3) *must* be properly filled in with an IP address.

Transmission Scripts and TCP/IP Connections

For TCP/IP connections, MailMan can now build transmission scripts on the fly. For transmission scripts whose TYPE is "SMTP," "TCPCHAN," or null, if the transmission script has no records (in the TEXT field [#2], in the TRANSMISSION SCRIPT multiple) in the DOMAIN file (#4.2), MailMan creates the script if the following new fields in the MAILMAN SITE PARAMETERS file (#4.3) are filled in:

- TCP/IP COMMUNICATIONS PROTOCOL (#8.23)—MailMan uses this field to determine which protocol shall be used for TCP/IP. It points to the COMMUNICATIONS PROTOCOL file (#3.4).

- TCP/IP TRANSMISSION SCRIPT (#8.24)—MailMan uses this field to determine which script shall be used for TCP/IP. It points to the TRANSMISSION SCRIPT file (#4.6).

Transmission Priority Flexibility

Messages in transmit queues can now be designated as low priority, as well as high priority. If a message gets stuck in a transmit queue and is holding up the rest of the queue for whatever reason, MailMan will make that message a low priority message, so that all the other messages are transmitted ahead of it. The Postmaster can also make these priority changes. In the message queue, high-priority messages are now marked with "^", instead of "$". Low priority messages are marked with "v". The Postmaster can now change the transmit priority at the message level (at the "Message action: Ignore//" prompt). As at the basket level, the command to use is "X".

CAUTION: In a user basket, the "X" at the message level is a command to unload a PackMan message or KIDS build. In a remote transmit queue, the "X" changes the transmit priority. The difference is the context, and writers of MailMan front-ends should take note

Reformatted Date/Times

MailMan date/times are now in a standard format, produced by Kernel's $$FMTE^XLFDT(datetime,"2Z") API. This call is a Supported IA. For example:

Date/Time Data	MailMan V. 7.1 Date/Time Format (Old)	MailMan V. 8.0 Date/Time Format (New)
3020803.153204	03 Aug 02 15:32	08/03/02@15:32

Table 1. MailMan V. 8.0: New date/time format

This change is also carried through to all MailMan APIs that return date/time in MailMan format.

REF: For more information on this Kernel API, please refer to the $$FMTE^XLFDT Home Page at the following Web address:

http://vista.med.va.gov/kernel/apis/x-fmte^xlfdt.asp

REF: For information on how to obtain a list of Integration Agreements (IAs) related to MailMan, please refer to the "Integration Agreements (IAs)" topic in the "External Relations" chapter in the *MailMan Technical Manual*.

Reformatted Remote Message IDs

MailMan remote message IDs now include the message date, to ensure that if you are told that a message is a duplicate of a previously received message, it really is. Sites will no longer have problems sending messages from a production account to a test account that was created by "mirroring" the production account. The remote message ID is now the message number followed by a period, followed by the 7-digit VA FileMan message creation date. For example:

MailMan V. 7.1 Remote Message ID (Old)	MailMan V. 8.0 Remote Message ID (New)
34561234@FORUM.VA.GOV	34561234.3020803@FORUM.VA.GOV

Table2. MailMan V. 8.0: New remote message ID format

Enhanced Routines

The ^XMC*, ^XMR*, ^XMS* suite of routines, which are responsible for scheduling, transmitting to, and receiving messages from remote sites, have been completely overhauled to make them easier to understand and easier to maintain.

Improved Name Field Display

MailMan will no longer display user names by taking them directly from the .01 field of the NEW PERSON file (#200). The Name Standardization $$NAMEFMT^XLFNAME API is used instead. Thus, the names of people whose last names, for example, contain periods, apostrophes, or spaces, are properly displayed. For example:

MailMan V. 7.1 Name Field Display (Old)	MailMan V. 8.0 Name Field Display (New)
STIVES	ST. IVES
OMALLEY	O'MALLEY
VANDYKE	VAN DYKE

Table 3. MailMan V. 8.0: New name field display format

Broadcast Messages with Responses

Messages with responses may no longer be forwarded to broadcast to all users. Such messages may have important information in the responses, and as we all know, responses are not auto-forwarded to remote

sites for users with auto-forward addresses. Users who attempt to broadcast messages with responses will be encouraged to copy the message and its responses into a new message, which can be broadcast.

Message Line Restriction Override

Incoming PackMan and KIDS messages are no longer subject to the restriction of the NETWORK - MAX LINES RECEIVE field (#8.31) in the MAILMAN SITE PARAMETERS file (#4.3). However, other kinds of messages continue to be subject to that restriction.

Requeued Task Bug Fix

If a task transmitting messages to another site fails and has to be requeued, it really is requeued. Before, that wasn't true. Previously, the failing task queued up a new task to take its place, and then the failing task stopped.