Mint Managers Manual

Manual last modified on 4 May 1989 at 15:57
by JDECK
Source is MGR-MANUAL.MSS, 03100/1
Comments to Joseph Deck (JDeck@Eagle.Wesleyan.Edu)

1. Installation

1.1. Disk space Requirements

The following sizes (in blocks) approximate the disk requirements to install and run MINT.

Disk space requirements				
	Base	Spell Check	Encrypt	Manual
Sources	2000	4200	250	2000
Executables	1500	4000	150	600

1.2. Installation Procedure

The installation of MINT has six steps. They are:

Step 1 restore the files from tape to disk

Step 2 execute the command procedure MINT-BUILD-STARTUP.COM

STEP 3 execute the command procedure MINT-BUILD-EXE.COM

STEP 4 execute the command procedure MINT-MOD-DCL.COM

STEP 5 modify the PMDF command procedure MASTER.COM

STEP 6 modify the template to create your system MAILING-LIST files

Note: To install MINT you must have previously installed PMDF. Read the section titled "MINT and PMDF" for more information on PMDF.

1.2.1. Step 1

To install MINT first decide if you want to install just the base MINT system or the base system plus other options such as the spelling checker, file encryption, etc. The tape has the following directory structure.

```
<Mint>----
          <encryption>
          <exe>
          <manual>----
                      <mgr>
                      <user>
          <object>
          <source>
          <spell>
```

Whether you choose to install just the base MINT system or the entire distribution you should maintain the directory structure that is on the tape by using the [...] syntax in the backup command. For example, the command would be similar to "*backup mua0:mint.bck [...]*" to restore the entire distribution to your default directory.

Alternately, you may choose to restore only the directories [.source] and [.object] which make up the base MINT system. If you do, remember to preserve the directory structure. With the files in these directories you can build the base MINT system.

1.2.2. Step 2

The command procedure MINT-BUILD-STARTUP.COM is used to create the command procedure file that will be used to install images and define system logical names. The created file, named MINT-STARTUP.COM, should be executed when the system is booted.

MINT-BUILD-STARTUP.COM prompts for the device and directories where the executable files and queued mail messages will be stored. Do not use a logical name for the device or the directory specification. It will request the internet host name and local domain name for the system. For more information on how these logical names are used read the section on logical names.

1.2.3. Step 3

The command procedure MINT-BUILD-EXE.COM if executed without arguments compiles, links, builds libraries, etc. If given the argument LINK it will skip the compilations. *If you don't have a C compiler you must use the LINK argument when executing this procedure.*

This command procedure will ask for a directory name where it expects to find two subdirectories, namely [.source] and [.object]. If you used the directory structure from the distribution the directory name will be [mint]. When this procedure has finished, all the files you will need to run MINT will be in the [.exe] subdirectory.

If this is a first time installation the files will be copied to the directory pointed to by logical name SYS$MINT. Otherwise you will be prompted as to whether or not to copy the files.

1.2.4. Step 4

At this point you should be able to run MINT. A sample message file must be copied to your SYS$LOGIN directory to simulate received mail. Copy the file MINT.TXT to your SYS$LOGIN directory. Type MINT at the command prompt to run MINT.

To make MINT available to all users you will need to modify your dcltables, install the images and define for the system some logical names.

To modify your DCLTABLES.EXE file execute the command procedure MINT-MOD-DCL.COM. This will add the commands MINT, MLIST, and PMDF to your system dcltables.

To install the images and define the required system logical names execute the command procedure created in step 2. MINT-STARTUP.COM should be executed now and added to your system startup command procedure.

1.2.5. Step 5

To have PMDF deliver mail to MINT and you must modify the PMDF file MASTER.COM and comment out the line " **mail/protocol=pmdf_mailshr 'file_to_process'** " and add the line " **pmdf 'file_to_process'** ". This will switch the local delivery of mail from VMS mail to MINT mail. The following shows you how to do this.

```
$!GR 14-APR-1989 Replace VAX MAIL processing with MINT
$!   mail/protocol=pmdf_mailshr 'file_to_process'
$ pmdf 'file_to_process'
```

1.2.6. Step 6

Edit the file SYS$MINT:MAILING-LIST.TXT to reflect your site. Then use the command MLIST/SYSTEM to create an indexed form of the file. For more information read the section titled "Mailing Lists" and refer to your PMDF documentation on alias files.

1.3. Logical Names

Five logical names are required to run MINT. They are: (SYS$MINT, SYS$MINTQ, INTERNET_HOST_NAME, SYS$MINT$DOMAIN, TPU$CALLUSER).

The logical name SYS$MINT is used to point to the directory where the executable, help, initialization, and text files are stored. This directory is protected so that it is world readable and executable.

The logical name SYS$MINTQ is used to point to the directory where the mail messages being queued to delivery system are stored. This directory is protected so that it is not accessible by world. MINT writes files into this directory using SYSPRV privilege. You should point this logical to the directory pointed to by PMDF_ROOT:[QUEUE]. This directory should be owned by an account that has a very high disk quota or be on a disk where quota checking is disabled.

The logical name INTERNET_HOST_NAME is used to specify the internet host name of the system on which MINT is being installed. An internet host name is made up of two parts; the host name and the domain name. For example, the host might be the system’s DECNET name and the domain name might be the name of the organization plus the appropiate upper level domain ending. Upper level domain ending include: MIL, COM, EDU, and ORG. An example of two INTERNET_HOST_NAMEs at Wesleyan are EAGLE.WESLEYAN.EDU and WREN.WESLEYAN.EDU where EAGLE and WREN are the machine’s local DECNET host names.

The logical name SYS$MINT$DOMAIN is used to restrict and disable the auto-reply feature of MINT. Should you want to disable auto-reply do not define this logical name. The SYS$MINT$DOMAIN logical could be set to be the INTERNET_HOST_NAME of the computer or just some part of the domain name. **Minimally you should restrict the auto-reply feature to work only in your own organization.**

If there were no restriction on the auto-reply feature there would be a potential for a mail loop with a mail system other than MINT. This would not occur in a MINT only environment. In a MINT only environment, should two users have auto-reply turned on and one of the users sends mail to the other, only one auto-reply would be generated for each user.

For example, if you had two systems, EAGLE.WESLEYAN.EDU and WREN.WESLEYAN.EDU and you wanted users to be able to use auto-reply across these two systems, define the logical to be the domain name. In this case define SYS$MINT$DOMAIN to be WESLEYAN.EDU. If you had wanted users on EAGLE to only be able to use auto-reply to other users on EAGLE define SYS$MINT$DOMAIN to be the same as the INTERNET_HOST_NAME "EAGLE.WESLEYAN.EDU".

The logical name TPU$CALLUSER is used to call the program WRTCOMMON.EXE from inside the TPU section file. WRTCOMMON.EXE is used to signal MINT from the tpu section file indicating that either the user wants to queue the message or exit to the send level of MINT.

1.4. Privileges

The program MINT needs to be installed with the following privileges to be run by non privileged users: CMKRNL, SYSLCK, SYSPRV, OPER, and EXQUOTA. The program CNTNEW.EXE needs to be installed with SYSLCK.

When MINT is first invoked CMKRNL is used to get the user process flags on what parts of messages should be displayed (facility, identification, severity, and text).

SYSLCK is used for locking of the message file so that an expunge of deleted messages will not interfere with delivery or reading of messages.

SYSPRV and OPER are used to write messages to the protected directory pointed to by the logical SYS$MINTQ during the process of queueing a message to mail delivery system and wakening the queueing mechanism.

EXQUOTA is used during the expunge process. The expunge subroutine creates a duplicate message file minus the messages that were marked for deletion. Expunging deleted messages could easily put a user over quota for a few seconds. EXQUOTA allows the user to exceed his disk quota for the duration of this process.

2. Programs and Files

2.1. Programs

2.1.1. MINT and PMDF

MINT.EXE is the user interface for the system. MINT.CLD is the corresponding CLD file.

A mail system requires a delivery agent as well as a user interface. MINT interfaces with PMDF which may be acquired from: The Clearinghouse for Academic Software, Computation Center, 104 Computer Science Building, Iowa State University, Ames, Iowa 50011: (515) 294-0323. The last time I checked it cost between $80 and $105 depending upon the distribution media.

NOTE: The line " mail/protocol=pmdf_mailshr 'file_to_process' " in the PMDF file MASTER.COM needs to be changed to "pmdf 'file_to_process' ". This will switch the local delivery of mail from VMS mail to MINT mail.

Also if you want to log local mail delivery add the following lines to MASTER.COM:

```
$!GR 14-APR-1989 Replace VAX MAIL processing with MINT
$!   mail/protocol=pmdf_mailshr 'file_to_process'
$!
$!GR 14-APR-1989
$!If you don't want to LOG the mail message in PMDF_ROOT:[LOG]MAIL.LOG
$! Then get rid of lines from here on till where you see
$!      close l_file.
$ ctrl_a[0,16] = %x0101
$ open/read m_file 'file_to_process'
$ open/append l_file pmdf_root:[log]mail.log
$ read/end=exit_this_loop/error=exit_this_loop m_file line_1
$ read/end=exit_this_loop/error=exit_this_loop m_file line_2
$ curr_time = f$time()
$ write l_file "Processed:",curr_time
$ write l_file line_1
$ write l_file line_2
$ read_more:
$ read/end=exit_this_loop/error=exit_this_loop m_file line_2
$ if (f$extract(0,2,line_2) .eqs. ctrl_a)  then goto exit_this_loop
$ write l_file line_2
$ goto read_more
$ exit_this_loop:
$ close m_file
$ close l_file
$! GR 14-APR-1989 Get rid of lines upto here.
$ pmdf 'file_to_process'
```

2.1.2. Spelling Checker

Mint invokes the spelling checker written by Mark Resmer at Vassar. The Spelling checker made publically available via DECUS library is installed separately from Mint. If you have a different spelling checker Mint will be able to use it if it can be invoked with the command line "spell filename".

2.1.3. New Mail Notification

The program CNTNEW can be put in the user's startup LOGIN.COM file or the system wide login command file SYSLOGIN.COM. This program counts the new (unread) messages. Mail is delivered via PMDF-DELIVER.EXE. If mail is delivered to a user while he/she is logged on a message is sent to his/her terminal.

2.1.4. TPU/ MINT communication

WRTCOMMON.EXE is used by the TPU section files to communicate with MINT by writing to the users common P1 area. This communication from the TPU section files directs MINT to either queue the message to PMDF or exit to the send level of MINT.

2.1.5. Encryption

Encryption is not currently available in MINT but will be included in a future release. Encryption routines are included in the distribution and may be installed after installing MINT.

2.1.6. Message Delivery

PMDF-DELIVER.EXE is the local delivery agent program which is called by PMDF. This program delivers messages to the users mail file. PMDF-DELIVER.CLD is the corresponding CLD file.

2.2. Files

2.2.1. Tpu Section Files

Mint by default uses a tpu section file to create messages. Mint can be configured to use the system editor. By assigning the symbol EDIT either at the user or system level the editor invoked by MINT can be changed.

There should be three tpu section files located in SYS$MINT: HCPY_EDITOR.TPU$SECTION (used for hard copy terminals), MINT-EDITOR.TPU$SECTION (used for normal users), and MINT-EDICAP.TPU$SECTION (used for captive accounts). Additionally, the file MAIL-TPU-HELP.TXT is the help file for the tpu section files.

2.2.2. Mailing Lists

There are two types of mailing lists available with MINT. One is maintained by the user as a personal list; the other is maintained by the system postmaster. Mint uses an indexed form of the mailing lists. The program MLIST.EXE is used to create the indexed form. The system mailing list text and index files must be located in the directory pointed to by the logical name SYS$MINT. An example of a system mailing list is supplied in the distribution. Use that as a template for your MAILING-LIST.TXT file.

Since pmdf also uses a mailing list concept (PMDF_ALIAS_FILE:) consideration should be given to consolidation for ease of maintenance. This can be done by pointing the PMDF logical PMDF_ALIAS_FILE: to the text form of the mailing list file. Another method of keeping these files in sync can be done by copying the file SYS$MINT:MAILING-LIST.TXT to PMDF_ALIAS_FILE. The command procedure named, MAILING-LKIST.COM can be used for this purpose.

2.2.3. Help

The MINT help files consist of TOPHELP.HLB, SENDHELP.HLB, READHELP.HLB, and POP.TXT. The first three files match the three levels of the MINT command structure. The last file is displayed when a control H (toggle) is hit displaying the commands available at the current command level.

3. Miscellaneous

3.1. Captive Accounts and UAF Flags

The user flag DISMAIL is checked and if set the user is not allowed to send mail. If the user has a mail file (mint.txt) he/she can read messages.

If the user flag CAPTIVE is set, commands that would allow the user to break out of MINT to the DCL level are disabled. Additionally, any command that reads in or writes out files is disabled. This was done under the assumption that a captive account would/could have privileges to give the account access to files required for the DCL command procedure, and that you would not want the account to have free access to the files. For example, if not disabled the copy command could copy a message overwriting the DCL command procedure which makes the captive account captive.

There are times, however, that the captive account would want to send a modified stock mail message. Therefore files located in the default directory of a captive account that begin with C_FORM_ can be read into a message being sent. For example, if the file C_FORM_LABELS was located in the default directory of a captive account the captive account could use the INSERT LABELS command of the tpu section to insert the file into the message being sent.

3.2. Hard Copy Terminals

MINT was written primarily to support vt100 compatible terminals. There is support for non vt100 compatible and hardcopy terminals, though it is primitive. To use MINT with one of these terminals use the DCL command "set terminal/noansi". This will use the hardcopy tpu section file when sending mail.

3.3. Disk Quotas

The mail delivery system disregards the user's disk quota when delivering mail. This was done because if a message is not delivered, PMDF will continually attempt to deliver the message until it is successful. To prevent this system overhead we decided to allow the delivery system (PMDF-DELIVER.EXE) to deliver the message regardless of the users disk quota.

The privilege EXQUOTA is also used in the user interface (MINT) to allow the user to expunge deleted messages. It is not used when sending mail. Since the send routines create a temporary file for use by the tpu section files, a user who is over quota can not send mail messages. This was our solution to entice users to get under disk quota without putting a burden on the message delivery system.

Should anyone wish to allow users to exceed quota when sending messages, in the file send.mar call the privon routine (from the ssend routine) and privoff routine (from the squit routine) with the exquota argument.

3.4. Registered Mail

A feature called registered mail is currently being added to MINT. This feature when completed will allow a user to specify that he/she wants to be notified when a message gets delivered locally or when it leaves the local system to be delivered some where else.

The support for Registered Mail has been completed in the MINT.EXE and the PMDF-DELIVER.EXE programs. To complete the Registered Mail process another program SEND.EXE is required. Currently the program has been completed for SRI's TCP/IP Multinet software. Prior to releasing it we will add support for DECNET.

Should you want to remove the Registered Mail command from MINT.EXE, edit the files TOP_COMMAND.CLD and SEND_COMMAND.CLD and remove the verbs and arguments that refer to register and unregister. You will then need to create new object files from them and relink MINT. To do this refer to the file MINT-BUILD-EXE.COM.

Table of Contents