Updating from Hermes 1.04 to Hermes 1.10

Martin P. Ibert

June 3, 1993

Contents

1. What is new?

Quite a number of minor improvements (?) will go unmentioned here, only a few major changes are documented.

1.1 MultiTOS and MiNT

Thanks to Robert Kießling, Hermes programs are now more compatible with MultiTOS and MiNT than they have previously been. But more work remains. Some of Robert's work did not work with plain MiNT, so I regrettably had to remove it after all.

1.2 `NEWS.DEF` format and location

The format of the `NEWS.DEF` file, which keeps track of known newsgroups and article numbers, has been changed to speed up news reading (more specifically, opening the news reader in GUS), and to increase compatibility with UNIX-derived software. Alas, there is also a drawback: If you call up a shell from GUS and receive more news (as a result from calling UUWORK, for example), the GUS news reader will possibly not recognize them until you quit and restart GUS.

The format used to list newsgroup names and highest article numbers, separated by a verticle bar character (|). This format has been completely changed. It now looks like a Cnews `/usr/lib/news/active` file. If the lower article number is given as 00000, then there are currently no articles in this group.

Since the name of the `NEWS.DEF` file can now be configured with a new configuration parameter, you can configure it to be whatever some UNIX-derived software considers its active file (!), and use that software and Hermes with no hassle whatsoever. E. g. to use NN, you don't need BUILDACT anymore if you tweak your installation a little.

The information in `NEWS.DEF` is maintained by RNEWS, EXPIRE, and NHM. Opening the GUS news reader is so much faster now because GUS does not have to scan each and every newsgroup folder to find the lowest available number anymore.

1.3 HISTORY location

The HISTORY file, previously located in the $NEWSDIR, can now be moved elsewhere. A new configuration parameter has been added to make this possible. Moving this file can make sense because it is a very volatile file and at least one user reports consistency problems with the hard disk partition that contains it. This is most probably not Hermes's fault, but since adding that parameter wasn't a big deal, I just did it.

1.4 Incoming mail storage

Incoming mail used to be stored as a collection of .MSG files in the user's home or MAIL directory. This has been changed. All mail messages for a user a now located in a single file (called a "mailbox" file). All mailbox files are located in a single directory, $MAILDIR. A configuration parameter has beed added to name the mailbox file directory.

If all mail messages are in a single file, there must be a way to tell where one message ends and the next begins. The "old" UNIX compatible way to tell is to scan that file for lines that begin with "From ". However, this is unacceptable because it is not transparent and very time-consuming.

Hermes[1] uses an upward-compatible way of separating the messages. While scanning the file, Hermes interprets the header lines of messages. If there is a "Content-Length:" header line, Hermes takes its value and skips that many bytes from the beginning of the message body without even looking at them. This has two advantages: It is *much* faster, and message bodies can contain lines beginning with "From " without confusing Hermes. Once it has skipped over the body (or right after the message header if no "Content-Length:" header line was found), Hermes starts looking for a "From " line. So Hermes can undestand old-style mailbox files as well. This appears to be compatible with the way elm (version 2.4), the popular UNIX mail program, handles mailbox files (if told to use that header line); Hermes can read elm mailbox files and vice versa. When writing back the mailbox file, Hermes adds "Content-Length:" header lines (and removes old ones, if any).

[1] In this context, "Hermes" is short for "any Hermes program that does what is just being described."

The MAIL.RC files are no longer used (only the timestamp is used to tell when the user last read his mail). Additional information is kept in an "X-Hermes-Info:" header line. It keeps track of new and read mail, as well as of the time when the message was sent and received. This is not exactly compatible with elm, but I don't expect users to use the same mailbox file alternately with elm and Hermes anyhow. At least, the way Hermes does it is more compatible with the relevant RFCs.

If you ever encounter a "mailbox locked" error, just remove the read-only attribute from the mailbox file.

Care must be taken if mailbox files are edited. Under no circumstances should mailbox files contain carriage-return (\r) characters. Okay, Hermes will, as a rule, read such mailbox files just fine, but don't push it. If you must edit a mailbox, use an editor that writes back files without carriage-return characters. Hermes has an imaginary boundary between the network and user domains. In the network domain, lines are \n-separated, and 8-bit characters are taken to be ISO-8859/1-coded. In the user domain, lines are \r\n-separated, and 8-bit characters are taken to be Atari-ST-coded. Mailbox files are, just as the .MSG files have been, in the network domain.

1.5 MIME support

Hermes now includes (limited) support for messages with RFC-1341-compatible header lines. The only header line that is interpreted is the "Content-Transfer-Encoding:" line. Only the "quoted-printable" encoding is supported[2].

When sending news or mail messages, the content transfer encoding can be chosen. "MIME-Version:", "Content-Type:" and "Content-Transfer-Encoding:" header lines are written as appropriate. Look at figure 1 for an example. This message has been sent from a Hermes system to a UNIX system running smail version 3.1.28 and elm version 2.4. The message body reads "'zwölf boxer jagen victor quer über den großen sylter deich". Hermes itself can read this gibberish because of the "Content-Transfer-Encoding: quoted-printable" header line. UNIX' elm can't, but it will call

[2]Even that is not completely supported, but I think the implementation is good enough.

metamail to do the job if so configured.

```
From cloud9.in-berlin.de!martini Wed Nov 25 13:46:27 1992
Return-Path: <martini@cloud9.in-berlin.de>
Received: by heaven7.in-berlin.de from cloud9.in-berlin.de with bsmtp
        (/\==/\ Smail3.1.28.1 #28.2)
        id <m0muM5W-000D0ZC>; Wed, 25 Nov 92 13:46 MET
Received: by cloud9.in-berlin.de (HERMES RMAIL 1.10 Rev. Nov 25 1992)
        id <0BY9VGG@cloud9.in-berlin.de>; 25 Nov 92 13:45 MET
From: martini@cloud9.in-berlin.de (Martin P. Ibert)
Message-ID: <H.eg.gPslgc2SaWo@cloud9.in-berlin.de>
Organization: The Seventh Heaven, Berlin, Germany
Subject: Test
To: martini@heaven7
X-Software: HERMES MAIL 1.10 Rev. Nov 25 1992
Date: Wed, 25 Nov 1992 13:45:42 MET
MIME-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable
Content-Length: 459
Status: RO

zw=F6lf boxer jagen victor quer =FCber den gro=DFen sylter deich
--
   __      | Martin P. Ibert, F=FCrstenweg 11, D-13589 Berlin (old: D-1000B20)
  (  )__   | martini@cloud9.in-berlin.de - Phone +4930-3753479, Fax -3753699
 (      )_ |-----------------------------------------------------------------
(_________) | All that we see or seem/is but a dream within a dream. --E.A.P.
```

Figure 1: Mail message with MIME header lines

2. How to update

2.1 Before you begin

Updating is neither foolproof nor exceptionally safe, so I recommend backing up all important files, especially those even remotely connected to Hermes, before you begin. Keep your old Hermes binaries in case you want, or have to, go back to version 1.04.

Of course, you were going to make a back-up anyway, weren't you?

Don't be afraid, updating does not require reformatting your hard disk or anything, I just don't want you to call me up and complain that the update trashed a year's worth of work. I don't know how it ever could, but then again, with computers, you just *never* know.

2.2 New binaries and documentation files

The new binaries are in the files `HERM110B.CXF` and `SUPPORT.CXF`. The new documentation files are in `HERM110D.CXF`. Look at the installation script `INSTALL.HSH` to see how to unpack them (using FREEZE and CPAR). All the files and programs you need are on the distribution disks.

2.3 The update program

There is a special utility program called 104TO110. It does most of what has to be done all by itself (if properly instructed).

There are three update phaes:

Phase 1 Adding the $MAILDIR, $HISTORYFILE, and $NEWSDEFFILE configuration parameters (lines 0121–0123) to the `CONFIG.SYS` file.

Phase 2 Rewriting the `NEWS.DEF` file.

Phase 3 Copying old mail files to mailbox files.

While 104TO110 does the first two phases pretty well, it doesn't do all that good a job with phase 3.

2.3.1 Running the update program

That is quite simple. Just double-click on it from the desktop, or run it from a shell. You have to tell it which phases you want; this is done with options: `-1` means phase 1, etc. If you want all three phases, say `104TO110 -123` to your shell, or give a parameter of `-123` to the desktop dialog box. If you run 104TO110 while a Hermes program is in memory (e. g., LOGIN or GUS), quit and restart all Hermes programs *immediately* after running 104TO110.

2.3.2 Phase 1

In phase 1, (ideally) three lines are appended to your `CONFIG.SYS` file. 104TO110 attempts to be clever about finding a good place for your mailbox files and will probably succeed in doing so, but if it fails, you will have to do it by hand. You can re-run phase 1 as often as necessary, or indeed as often as you want (provided you do it as described in 2.3.1), because then it will not append those lines more than once.

You can change the lines 104TO110 added if you like, but be sure to move the stuff they refer to around as well. E. g. you might change line 0123 to be whatever your other software considers its `active` file, but if you do, you have to move the Hermes `NEWS.DEF` file to that new location yourself.

2.3.3 Phase 2

In phase 2, your `NEWS.DEF` file is read (in either the old or the new format) and written back in the new format. This means you can do that as often as you like, but doing it more often than once will just take time and do no good (but no harm either).

2.3.4 Phase 3

In phase 3, all users (as listed in the `PASSWD` file) will get their `.MSG` files concatenated to a mailbox file. This will not work unless phase 1 (or the manual equivalent) has been completed first. The files are just concatenated; no new header lines are added, and the information in the

MAIL.RC is lost. (This is because I was too lazy to do it properly. It is possible, but I don't think it is worth the effort.)

As a consequence, mail messages with "From " lines in the body will be split up into pieces! Fortunately, such messages are rare. As another consequence, opening the new mailbox files for the first time will take forever. It will be *much* faster the second time, or indeed any time after the first—but only if you do something to your mailbox the first time you open it. Just delete any message, or mark a read message as unread or vice versa. Otherwise, your mailbox file will not be rewritten.

2.4 Cleaning up

- Look at your CONFIG.SYS file. Are the new lines as you want them? If not, change them, but remember to move any stuff they might refer to in accordance to their new values.
- Open your mailbox file (with MAIL or GUS). Is everything alright? Do something to it, and close it. If all went well, remove your .MSG files and your $HOME\MAIL folder (if any). Truncate your MAIL.RC file to 0 bytes or remove it.

  This should be done once by each user.
- Look at the $MAILDIR folder. There are probably some empty mailbox files. You can safely remove them.