

NAME

MMDF – Multi-channel Memorandum Distribution Facility mailbox format

DESCRIPTION

This document describes the **MMDF** mailbox format used by some MTAs and MUAs (i.e. **scomail(1)**) to store mail messages locally.

An **MMDF** mailbox is a text file containing an arbitrary number of e-mail messages. Each message consists of a postmark, followed by an e-mail message formatted according to **RFC822** / **RFC2822**, followed by a postmark. The file format is line-oriented. Lines are separated by line feed characters (ASCII 10). A postmark line consists of the four characters "**^A^A^A^A**" (Control-A; ASCII 1).

Example of a **MMDF** mailbox holding two mails:

```

^A^A^A^A
From: example@example.com
To: example@example.org
Subject: test

>From what I learned about the MMDF-format:
^A^A^A^A
^A^A^A^A
From: example@example.com
To: example@example.org
Subject: test 2

bar
^A^A^A^A

```

In contrast to most other single file mailbox formats like MBOXO and MBOXRD (see **mbox(5)**) there is no need to quote/dequote "From"–lines in **MMDF** mailboxes as such lines have no special meaning in this format.

If the modification-time (usually determined via **stat(2)**) of a nonempty mailbox file is greater than the access-time the file has new mail. Many MUAs place a Status: header in each message to indicate which messages have already been read.

LOCKING

Since **MMDF** files are frequently accessed by multiple programs in parallel, **MMDF** files should generally not be accessed without locking.

Three different locking mechanisms (and combinations thereof) are in general use:

- **fcntl(2)** locking is mostly used on recent, POSIX-compliant systems. Use of this locking method is, in particular, advisable if **MMDF** files are accessed through the Network File System (NFS), since it seems the only way to reliably invalidate NFS clients' caches.
- **flock(2)** locking is mostly used on BSD-based systems.
- Dotlocking is used on all kinds of systems. In order to lock an **MMDF** file named *folder*, an application first creates a temporary file with a unique name in the directory in which the *folder* resides. The application then tries to use the **link(2)** system call to create a hard link named *folder.lock* to the temporary file. The success of the **link(2)** system call should be additionally verified using **stat(2)** calls. If the link has succeeded, the mail folder is considered dotlocked. The temporary file can then safely be unlinked.

In order to release the lock, an application just unlinks the *folder.lock* file.

If multiple methods are combined, implementors should make sure to use the non-blocking variants of the **fcntl(2)** and **flock(2)** system calls in order to avoid deadlocks.

If multiple methods are combined, an **MMDF** file must not be considered to have been successfully locked

before all individual locks were obtained. When one of the individual locking methods fails, an application should release all locks it acquired successfully, and restart the entire locking procedure from the beginning, after a suitable delay.

The locking mechanism used on a particular system is a matter of local policy, and should be consistently used by all applications installed on the system which access **MMDF** files. Failure to do so may result in loss of e-mail data, and in corrupted **MMDF** files.

CONFORMING TO

MMDF is not part of any currently supported standard.

HISTORY

MMDF was developed at the University of Delaware by Dave Crocker.

SEE ALSO

scomail(1), fcntl(2), flock(2), link(2), stat(2), mbox(5), RFC822, RFC2822

AUTHOR

Urs Janssen <urs@tin.org>