

# NeXT mail index file format

*What follows is preliminary information I collected by playing around with the NeXT Mail table\_of\_contents files that are a part of all NeXT Mail mailboxes. Please send comments or improvements to cap+@cmu.edu.*

Index files consist of a 32 byte header followed by some number of variable length records containing information about each message in the mailbox. The number of these records is equal to the number of messages in the mailbox. We can describe the header with the following C structure. I'm using a C structure only for clarity, and I'm assuming that there is no padding between structure elements. That is, in the index files, each field takes up exactly as much space as the corresponding field in the C structure.

```
struct table_of_contents_header {
    long magic;           /* magic number: 0xd9758 */
    long num_msgs;       /* number of messages in mbox */
    long mbox_time;     /* the m_time of mbox */
    long mystery[5];    /* I don't know what these are for */
};
```

All numbers appear in big-endian format. The **magic** field holds the value 0xd9758 in every example I looked at. It's reasonable to think it is a constant, though I cannot be sure. The **num\_msgs** field is the number of messages in the mailbox's **mbox** file. The **mbox\_time** field is the last modification time of **mbox**. This is the **st\_mtime** field of **struct stat**. If, when Mail opens a mailbox, the **mbox\_time** field of the **table\_of\_contents** file does not match the modification time of **mbox**, Mail will rebuild the **table\_of\_contents** file, and will assume that all messages have been read. The remainder of the

header consists of five long words that I've marked **mystery**, because I can't figure out what they do. Mail does set them to different values, but it doesn't seem to care what values they have. I have a program that writes `table_of_contents` files, and I write the **mystery** information as all 0's. Perhaps they're reserved for future use.

Immediately after the header, index files contain a set of records, one per message. The records consist of a fixed-length section followed by three null-terminated strings. We can describe the fixed length section by the following structure:

```
struct message_index {
    long record_length;      /* the length of this record, including
this word */
    long message_offset;    /* offset in mbox where this message
```

```
begins */
    long message_length;    /* length in bytes of this message in
mbox */
    long message_date;      /* the date of this message */
    char status;           /* read, unread, deleted */
    char msgtype;          /* regular or NeXT mail */
    char mystery[2];       /* I don't know what these are */
};
```

The **record\_length** field indicates the length of this record, including the space taken by the **record\_length** field and the space taken by the strings which appear after the above structure. The **message\_offset** is the byte offset into the mbox file where the first character of the message is located. The **message\_length** is the length, in bytes, of the message. The **message\_date** field holds the month, date, and year of the message, using the following

encoding:

bits 9-31: the year

bits 5-8: the month of the year (1 = January)

bits 0-4: the date of the month

The **status** field indicates things like whether the user has read or deleted this message. A value of `d` indicates the user has deleted this message (but not compacted the mailbox), and `\*` indicates that the user has not read the message. For messages that the user has read but not deleted, the status value is either ` ` (a space character), or `>`. I don't know the distinction between these two. The **msgtype** field contains `r` if the message is NeXT mail, or ` ` (space) otherwise. The fields I've marked **mystery** have contained space characters in all the examples I saw.

Following the static fields I just described, there are three strings, each terminated by a null character. The first is the contents of the message's `From:` header, excluding the `From:.` The second is the contents of the message's `Subject:` line, excluding the `Subject:.` If the message does not contain either of these three headers, the corresponding string in the `table_of_contents` file will be empty. The null terminator still present though. The third string is the name of the directory containing any NeXT mail for this message. This string is empty for non-NeXT mail.

Let's look at what these strings would look like for a NeXT mail message containing the header lines:

```
Next-Reference: Getting_Closer____.attach, 1/1
From: joe@hollowood.andrew.cmu.edu (Joseph Hollowood)
Subject: Getting Closer...
```

The strings have the form:

```
joe@hollowood.andrew.cmu.edu (Joseph Hollowood)\000Getting  
Closer...\000Getting_Closer____.attach
```

where \000 represents the ASCII null character.

The above information is obviously incomplete, so if you uncover any errors, or any information about the **mystery** fields, please let me know by sending mail to me at:

Chris Paris  
cap+@cmu.edu