:mod:`mailbox` --- Manipulate mailboxes in various formats

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 1); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 4)
>
> Unknown directive type "module".
>
> ```
>     .. module:: mailbox
>        :synopsis: Manipulate mailboxes in various formats
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 7)
>
> Unknown directive type "moduleauthor".
>
> ```
>     .. moduleauthor:: Gregory K. Johnson <gkj@gregorykjohnson.com>
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 8)
>
> Unknown directive type "sectionauthor".
>
> ```
>     .. sectionauthor:: Gregory K. Johnson <gkj@gregorykjohnson.com>
> ```

Source code: :source:`Lib/mailbox.py`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 10); *backlink*
>
> Unknown interpreted text role "source".

This module defines two classes, :class:`Mailbox` and :class:`Message`, for accessing and manipulating on-disk mailboxes and the messages they contain. :class:`Mailbox` offers a dictionary-like mapping from keys to messages. :class:`Message` extends the :mod:`email.message` module's :class:`~email.message.Message` class with format-specific state and behavior. Supported mailbox formats are Maildir, mbox, MH, Babyl, and MMDF.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 14); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 22)
>
> Unknown directive type "seealso".
>
> ```
>     .. seealso::
>
>        Module :mod:`email`
>           Represent and manipulate messages.
> ```

:class:`Mailbox` objects

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 30); *backlink*

Unknown interpreted text role "class".

A mailbox, which may be inspected and modified.

The :class:`Mailbox` class defines an interface and is not intended to be instantiated. Instead, format-specific subclasses should inherit from :class:`Mailbox` and your code should instantiate a particular subclass.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 37`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 37`); *backlink*
>
> Unknown interpreted text role "class".

The :class:`Mailbox` interface is dictionary-like, with small keys corresponding to messages. Keys are issued by the :class:`Mailbox` instance with which they will be used and are only meaningful to that :class:`Mailbox` instance. A key continues to identify a message even if the corresponding message is modified, such as by replacing it with another message.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 41`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 41`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 41`); *backlink*
>
> Unknown interpreted text role "class".

Messages may be added to a :class:`Mailbox` instance using the set-like method :meth:`add` and removed using a `del` statement or the set-like methods :meth:`remove` and :meth:`discard`.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 47`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 47`); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 47`); *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 47`); *backlink*
>
> Unknown interpreted text role "meth".

:class:`Mailbox` interface semantics differ from dictionary semantics in some noteworthy ways. Each time a message is requested, a new representation (typically a :class:`Message` instance) is generated based upon the current state of the mailbox. Similarly, when a message is added to a :class:`Mailbox` instance, the provided message representation's contents are copied. In neither case is a reference to the message representation kept by the :class:`Mailbox` instance.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 51`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 51`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 51`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 51`); *backlink*
>
> Unknown interpreted text role "class".

The default :class:`Mailbox` iterator iterates over message representations, not keys as the default dictionary iterator does. Moreover, modification of a mailbox during iteration is safe and well-defined. Messages added to the mailbox after an iterator is created will not be seen by the iterator. Messages removed from the mailbox before the iterator yields them will be silently skipped, though using a key from an iterator may result in a :exc:`KeyError` exception if the corresponding message is subsequently removed.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 59`); *backlink*
>
> Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 59); *backlink*

Unknown interpreted text role "exc".

Warning

Be very cautious when modifying mailboxes that might be simultaneously changed by some other process. The safest mailbox format to use for such tasks is Maildir; try to avoid using single-file formats such as mbox for concurrent writing. If you're modifying a mailbox, you *must* lock it by calling the :meth:`lock` and :meth:`unlock` methods *before* reading any messages in the file or making any changes by adding or deleting a message. Failing to lock the mailbox runs the risk of losing messages or corrupting the entire mailbox.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 70);** *backlink*
>
> Unknown interpreted text role "meth".

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 70);** *backlink*
>
> Unknown interpreted text role "meth".

:class:`Mailbox` instances have the following methods:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 79); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 82)

Unknown directive type "method".

```
.. method:: add(message)

   Add *message* to the mailbox and return the key that has been assigned to
   it.

   Parameter *message* may be a :class:`Message` instance, an
   :class:`email.message.Message` instance, a string, a byte string, or a
   file-like object (which should be open in binary mode). If *message* is
   an instance of the
   appropriate format-specific :class:`Message` subclass (e.g., if it's an
   :class:`mboxMessage` instance and this is an :class:`mbox` instance), its
   format-specific information is used. Otherwise, reasonable defaults for
   format-specific information are used.

   .. versionchanged:: 3.2
      Support for binary input was added.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 100)

Unknown directive type "method".

```
.. method:: remove(key)
            __delitem__(key)
            discard(key)

   Delete the message corresponding to *key* from the mailbox.

   If no such message exists, a :exc:`KeyError` exception is raised if the
   method was called as :meth:`remove` or :meth:`__delitem__` but no
   exception is raised if the method was called as :meth:`discard`. The
   behavior of :meth:`discard` may be preferred if the underlying mailbox
   format supports concurrent modification by other processes.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 113)

Unknown directive type "method".

```
.. method:: __setitem__(key, message)

   Replace the message corresponding to *key* with *message*. Raise a
   :exc:`KeyError` exception if no message already corresponds to *key*.

   As with :meth:`add`, parameter *message* may be a :class:`Message`
   instance, an :class:`email.message.Message` instance, a string, a byte
   string, or a file-like object (which should be open in binary mode). If
   *message* is an
   instance of the appropriate format-specific :class:`Message` subclass
   (e.g., if it's an :class:`mboxMessage` instance and this is an
   :class:`mbox` instance), its format-specific information is
   used. Otherwise, the format-specific information of the message that
   currently corresponds to *key* is left unchanged.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 129)

Unknown directive type "method".

```
.. method:: iterkeys()
            keys()

    Return an iterator over all keys if called as :meth:`iterkeys` or return a
    list of keys if called as :meth:`keys`.
```

Unknown directive type "method".

```
.. method:: itervalues()
            __iter__()
            values()

    Return an iterator over representations of all messages if called as
    :meth:`itervalues` or :meth:`__iter__` or return a list of such
    representations if called as :meth:`values`. The messages are represented
    as instances of the appropriate format-specific :class:`Message` subclass
    unless a custom message factory was specified when the :class:`Mailbox`
    instance was initialized.

    .. note::

        The behavior of :meth:`__iter__` is unlike that of dictionaries, which
        iterate over keys.
```

Unknown directive type "method".

```
.. method:: iteritems()
            items()

    Return an iterator over (*key*, *message*) pairs, where *key* is a key and
    *message* is a message representation, if called as :meth:`iteritems` or
    return a list of such pairs if called as :meth:`items`. The messages are
    represented as instances of the appropriate format-specific
    :class:`Message` subclass unless a custom message factory was specified
    when the :class:`Mailbox` instance was initialized.
```

Unknown directive type "method".

```
.. method:: get(key, default=None)
            __getitem__(key)

    Return a representation of the message corresponding to *key*. If no such
    message exists, *default* is returned if the method was called as
    :meth:`get` and a :exc:`KeyError` exception is raised if the method was
    called as :meth:`__getitem__`. The message is represented as an instance
    of the appropriate format-specific :class:`Message` subclass unless a
    custom message factory was specified when the :class:`Mailbox` instance
    was initialized.
```

Unknown directive type "method".

```
.. method:: get_message(key)

    Return a representation of the message corresponding to *key* as an
    instance of the appropriate format-specific :class:`Message` subclass, or
    raise a :exc:`KeyError` exception if no such message exists.
```

Unknown directive type "method".

```
.. method:: get_bytes(key)

    Return a byte representation of the message corresponding to *key*, or
    raise a :exc:`KeyError` exception if no such message exists.

    .. versionadded:: 3.2
```

Unknown directive type "method".

```
.. method:: get_string(key)

    Return a string representation of the message corresponding to *key*, or
    raise a :exc:`KeyError` exception if no such message exists.  The
    message is processed through :class:`email.message.Message` to
    convert it to a 7bit clean representation.
```

Unknown directive type "method".

```
.. method:: get_file(key)

   Return a file-like representation of the message corresponding to *key*,
   or raise a :exc:`KeyError` exception if no such message exists.  The
   file-like object behaves as if open in binary mode.  This file should be
   closed once it is no longer needed.

   .. versionchanged:: 3.2
      The file object really is a binary file; previously it was incorrectly
      returned in text mode.  Also, the file-like object now supports the
      context management protocol: you can use a :keyword:`with` statement to
      automatically close it.

   .. note::

      Unlike other representations of messages, file-like representations are
      not necessarily independent of the :class:`Mailbox` instance that
      created them or of the underlying mailbox.  More specific documentation
      is provided by each subclass.
```

Unknown directive type "method".

```
.. method:: __contains__(key)

   Return ``True`` if *key* corresponds to a message, ``False`` otherwise.
```

Unknown directive type "method".

```
.. method:: __len__()

   Return a count of messages in the mailbox.
```

Unknown directive type "method".

```
.. method:: clear()

   Delete all messages from the mailbox.
```

Unknown directive type "method".

```
.. method:: pop(key, default=None)

   Return a representation of the message corresponding to *key* and delete
   the message. If no such message exists, return *default*. The message is
   represented as an instance of the appropriate format-specific
   :class:`Message` subclass unless a custom message factory was specified
   when the :class:`Mailbox` instance was initialized.
```

Unknown directive type "method".

```
.. method:: popitem()

   Return an arbitrary (*key*, *message*) pair, where *key* is a key and
   *message* is a message representation, and delete the corresponding
   message. If the mailbox is empty, raise a :exc:`KeyError` exception. The
   message is represented as an instance of the appropriate format-specific
   :class:`Message` subclass unless a custom message factory was specified
   when the :class:`Mailbox` instance was initialized.
```

Unknown directive type "method".

```
.. method:: update(arg)

   Parameter *arg* should be a *key*-to-*message* mapping or an iterable of
   (*key*, *message*) pairs. Updates the mailbox so that, for each given
   *key* and *message*, the message corresponding to *key* is set to
   *message* as if by using :meth:`__setitem__`. As with :meth:`__setitem__`,
   each *key* must already correspond to a message in the mailbox or else a
   :exc:`KeyError` exception will be raised, so in general it is incorrect
   for *arg* to be a :class:`Mailbox` instance.

   .. note::

      Unlike with dictionaries, keyword arguments are not supported.
```

Unlike other representations of messages, file-like representations are

:class:`Maildir`

A subclass of :class:`Mailbox` for mailboxes in Maildir format. Parameter *factory* is a callable object that accepts a file-like message representation (which behaves as if opened in binary mode) and returns a custom representation. If *factory* is None, :class:`MaildirMessage` is used as the default message representation. If *create* is True, the mailbox is created if it does not exist.

If *create* is True and the *dirname* path exists, it will be treated as an existing maildir without attempting to verify its directory layout.

It is for historical reasons that *dirname* is named as such rather than *path*.

Maildir is a directory-based mailbox format invented for the qmail mail transfer agent and now widely supported by other programs. Messages in a Maildir mailbox are stored in separate files within a common directory structure. This design allows Maildir mailboxes to be accessed and modified by multiple unrelated programs without data corruption, so file locking is unnecessary.

Maildir mailboxes contain three subdirectories, namely: :file:`tmp`, :file:`new`, and :file:`cur`. Messages are created momentarily in the :file:`tmp` subdirectory and then moved to the :file:`new` subdirectory to finalize delivery. A mail user agent may subsequently move the message to the :file:`cur` subdirectory and store information about the state of the message in a special "info" section appended to its file name.

Folders of the style introduced by the Courier mail transfer agent are also supported. Any subdirectory of the main mailbox is considered a folder if `'.'` is the first character in its name. Folder names are represented by :class:`Maildir` without the leading `'.'`. Each folder is itself a Maildir mailbox but should not contain other folders. Instead, a logical nesting is indicated using `'.'` to delimit levels, e.g., "Archived.2005.07".

> **Note**
>
> The Maildir specification requires the use of a colon (`':'`) in certain message file names. However, some operating systems do not permit this character in file names, If you wish to use a Maildir-like format on such an operating system, you should specify another character to use instead. The exclamation point (`'!'`) is a popular choice. For example:
>
> ```
> import mailbox
> mailbox.Maildir.colon = '!'
> ```
>
> The :attr:`colon` attribute may also be set on a per-instance basis.
>

:class:`Maildir` instances have all of the methods of :class:`Mailbox` in addition to the following:

```
.. method:: list_folders()

    Return a list of the names of all folders.
```

```
.. method:: get_folder(folder)

    Return a :class:`Maildir` instance representing the folder whose name is
    *folder*. A :exc:`NoSuchMailboxError` exception is raised if the folder
    does not exist.
```

```
.. method:: add_folder(folder)

    Create a folder whose name is *folder* and return a :class:`Maildir`
    instance representing it.
```

```
.. method:: remove_folder(folder)

    Delete the folder whose name is *folder*. If the folder contains any
    messages, a :exc:`NotEmptyError` exception will be raised and the folder
```

```
       will not be deleted.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 380)

Unknown directive type "method".

```
.. method:: clean()

   Delete temporary files from the mailbox that have not been accessed in the
   last 36 hours. The Maildir specification says that mail-reading programs
   should do this occasionally.
```

Some :class:`Mailbox` methods implemented by :class:`Maildir` deserve special remarks:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 386); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 386); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 390)

Unknown directive type "method".

```
.. method:: add(message)
            __setitem__(key, message)
            update(arg)

   .. warning::

      These methods generate unique file names based upon the current process
      ID. When using multiple threads, undetected name clashes may occur and
      cause corruption of the mailbox unless threads are coordinated to avoid
      using these methods to manipulate the same mailbox simultaneously.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 402)

Unknown directive type "method".

```
.. method:: flush()

   All changes to Maildir mailboxes are immediately applied, so this method
   does nothing.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 408)

Unknown directive type "method".

```
.. method:: lock()
            unlock()

   Maildir mailboxes do not support (or require) locking, so these methods do
   nothing.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 415)

Unknown directive type "method".

```
.. method:: close()

   :class:`Maildir` instances do not keep any open files and the underlying
   mailboxes do not support locking, so this method does nothing.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 421)

Unknown directive type "method".

```
.. method:: get_file(key)

   Depending upon the host platform, it may not be possible to modify or
   remove the underlying message while the returned file remains open.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, line 427)

Unknown directive type "seealso".

```
.. seealso::

   `maildir man page from Courier <http://www.courier-mta.org/maildir.html>`_
      A specification of the format. Describes a common extension for
      supporting folders.

   `Using maildir format <https://cr.yp.to/proto/maildir.html>`_
```

```
      Notes on Maildir by its inventor. Includes an updated name-creation scheme and
      details on "info" semantics.
```

:class:`mbox`

A subclass of :class:`Mailbox` for mailboxes in mbox format. Parameter *factory* is a callable object that accepts a file-like message representation (which behaves as if opened in binary mode) and returns a custom representation. If *factory* is `None`, :class:`mboxMessage` is used as the default message representation. If *create* is `True`, the mailbox is created if it does not exist.

The mbox format is the classic format for storing mail on Unix systems. All messages in an mbox mailbox are stored in a single file with the beginning of each message indicated by a line whose first five characters are "From ".

Several variations of the mbox format exist to address perceived shortcomings in the original. In the interest of compatibility, :class:`mbox` implements the original format, which is sometimes referred to as :dfn:`mboxo`. This means that the :mailheader:`Content-Length` header, if present, is ignored and that any occurrences of "From " at the beginning of a line in a message body are transformed to ">From " when storing the message, although occurrences of ">From " are not transformed to "From " when reading the message.

Some :class:`Mailbox` methods implemented by :class:`mbox` deserve special remarks:

```
.. method:: get_file(key)

    Using the file after calling :meth:`flush` or :meth:`close` on the
    :class:`mbox` instance may yield unpredictable results or raise an
    exception.
```

```
.. method:: lock()
            unlock()

    Three locking mechanisms are used---dot locking and, if available, the
    :c:func:`flock` and :c:func:`lockf` system calls.
```

```
.. seealso::

    `mbox man page from tin <http://www.tin.org/bin/man.cgi?section=5&topic=mbox>`_
        A specification of the format, with details on locking.

    `Configuring Netscape Mail on Unix: Why The Content-Length Format is Bad <https://www.jwz.org/doc/content-length.html>`
        An argument for using the original mbox format rather than a variation.
```

```
      `"mbox" is a family of several mutually incompatible mailbox formats <https://www.loc.gov/preservation/digital/formats
          A history of mbox variations.
```

:class:`MH`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 497**); *backlink*
>
> Unknown interpreted text role "class".

A subclass of :class:`Mailbox` for mailboxes in MH format. Parameter *factory* is a callable object that accepts a file-like message representation (which behaves as if opened in binary mode) and returns a custom representation. If *factory* is `None`, :class:`MHMessage` is used as the default message representation. If *create* is `True`, the mailbox is created if it does not exist.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 503**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 503**); *backlink*
>
> Unknown interpreted text role "class".

MH is a directory-based mailbox format invented for the MH Message Handling System, a mail user agent. Each message in an MH mailbox resides in its own file. An MH mailbox may contain other MH mailboxes (called :dfn:`folders`) in addition to messages. Folders may be nested indefinitely. MH mailboxes also support :dfn:`sequences`, which are named lists used to logically group messages without moving them to sub-folders. Sequences are defined in a file called :file:`.mh_sequences` in each folder.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 510**); *backlink*
>
> Unknown interpreted text role "dfn".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 510**); *backlink*
>
> Unknown interpreted text role "dfn".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 510**); *backlink*
>
> Unknown interpreted text role "file".

The :class:`MH` class manipulates MH mailboxes, but it does not attempt to emulate all of :program:`mh`'s behaviors. In particular, it does not modify and is not affected by the :file:`context` or :file:`.mh_profile` files that are used by :program:`mh` to store its state and configuration.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "program".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "file".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "file".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 518**); *backlink*
>
> Unknown interpreted text role "program".

:class:`MH` instances have all of the methods of :class:`Mailbox` in addition to the following:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 523**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 523**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 527**)
>
> Unknown directive type "method".
>
> ```
>    .. method:: list_folders()
> ```

```
Return a list of the names of all folders.
```

```
.. method:: get_folder(folder)

   Return an :class:`MH` instance representing the folder whose name is
   *folder*. A :exc:`NoSuchMailboxError` exception is raised if the folder
   does not exist.
```

```
.. method:: add_folder(folder)

   Create a folder whose name is *folder* and return an :class:`MH` instance
   representing it.
```

```
.. method:: remove_folder(folder)

   Delete the folder whose name is *folder*. If the folder contains any
   messages, a :exc:`NotEmptyError` exception will be raised and the folder
   will not be deleted.
```

```
.. method:: get_sequences()

   Return a dictionary of sequence names mapped to key lists. If there are no
   sequences, the empty dictionary is returned.
```

```
.. method:: set_sequences(sequences)

   Re-define the sequences that exist in the mailbox based upon *sequences*,
   a dictionary of names mapped to key lists, like returned by
   :meth:`get_sequences`.
```

```
.. method:: pack()

   Rename messages in the mailbox as necessary to eliminate gaps in
   numbering.  Entries in the sequences list are updated correspondingly.

   .. note::

      Already-issued keys are invalidated by this operation and should not be
      subsequently used.
```

Some :class:`Mailbox` methods implemented by :class:`MH` deserve special remarks:

```
.. method:: remove(key)
            __delitem__(key)
            discard(key)

   These methods immediately delete the message. The MH convention of marking
```

```
                 a message for deletion by prepending a comma to its name is not used.
```

```
    .. method:: lock()
                unlock()

       Three locking mechanisms are used---dot locking and, if available, the
       :c:func:`flock` and :c:func:`lockf` system calls. For MH mailboxes, locking
       the mailbox means locking the :file:`.mh_sequences` file and, only for the
       duration of any operations that affect them, locking individual message
       files.
```

```
    .. method:: get_file(key)

       Depending upon the host platform, it may not be possible to remove the
       underlying message while the returned file remains open.
```

```
    .. method:: flush()

       All changes to MH mailboxes are immediately applied, so this method does
       nothing.
```

```
    .. method:: close()

       :class:`MH` instances do not keep any open files, so this method is
       equivalent to :meth:`unlock`.
```

```
    .. seealso::

       `nmh - Message Handling System <http://www.nongnu.org/nmh/>`_
          Home page of :program:`nmh`, an updated version of the original :program:`mh`.

       `MH & nmh: Email for Users & Programmers <https://rand-mh.sourceforge.io/book/>`_
          A GPL-licensed book on :program:`mh` and :program:`nmh`, with some information
          on the mailbox format.
```

:class:`Babyl`

A subclass of :class:`Mailbox` for mailboxes in Babyl format. Parameter *factory* is a callable object that accepts a file-like message representation (which behaves as if opened in binary mode) and returns a custom representation. If *factory* is None, :class:`BabylMessage` is used as the default message representation. If *create* is `True`, the mailbox is created if it does not exist.

Babyl is a single-file mailbox format used by the Rmail mail user agent included with Emacs. The beginning of a message is indicated by a line containing the two characters Control-Underscore (`'\037'`) and Control-L (`'\014'`). The end of a message is indicated by the start of the next message or, in the case of the last message, a line containing a Control-Underscore (`'\037'`) character.

Messages in a Babyl mailbox have two sets of headers, original headers and so-called visible headers. Visible headers are typically a subset of the original headers that have been reformatted or abridged to be more attractive. Each message in a Babyl mailbox also has an accompanying list of :dfn:`labels`, or short strings that record extra information about the message, and a list of all user-defined labels found in the mailbox is kept in the Babyl options section.

Unknown interpreted text role "dfn".

:class:`Babyl` instances have all of the methods of :class:`Mailbox` in addition to the following:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 655**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 655**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 659**)

Unknown directive type "method".

```
.. method:: get_labels()

    Return a list of the names of all user-defined labels used in the mailbox.

    .. note::

        The actual messages are inspected to determine which labels exist in
        the mailbox rather than consulting the list of labels in the Babyl
        options section, but the Babyl section is updated whenever the mailbox
        is modified.
```

Some :class:`Mailbox` methods implemented by :class:`Babyl` deserve special remarks:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 670**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 670**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 674**)

Unknown directive type "method".

```
.. method:: get_file(key)

    In Babyl mailboxes, the headers of a message are not stored contiguously
    with the body of the message. To generate a file-like representation, the
    headers and body are copied together into an :class:`io.BytesIO` instance,
    which has an API identical to that of a
    file. As a result, the file-like object is truly independent of the
    underlying mailbox but does not save memory compared to a string
    representation.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 685**)

Unknown directive type "method".

```
.. method:: lock()
            unlock()

    Three locking mechanisms are used---dot locking and, if available, the
    :c:func:`flock` and :c:func:`lockf` system calls.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 692**)

Unknown directive type "seealso".

```
.. seealso::

    `Format of Version 5 Babyl Files <https://quimby.gnus.org/notes/BABYL>`_
        A specification of the Babyl format.

    `Reading Mail with Rmail <https://www.gnu.org/software/emacs/manual/html_node/emacs/Rmail.html>`_
        The Rmail manual, with some information on Babyl semantics.
```

:class:`MMDF`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 703**); *backlink*

Unknown interpreted text role "class".

A subclass of :class:`Mailbox` for mailboxes in MMDF format. Parameter *factory* is a callable object that accepts a file-like message representation (which behaves as if opened in binary mode) and returns a custom representation. If *factory* is None, :class:`MMDFMessage` is used as the default message representation. If *create* is True, the mailbox is created if it does not exist.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 709**); *backlink*

Unknown interpreted text role "class".

MMDF is a single-file mailbox format invented for the Multichannel Memorandum Distribution Facility, a mail transfer agent. Each message is in the same form as an mbox message but is bracketed before and after by lines containing four Control-A (`'\001'`) characters. As with the mbox format, the beginning of each message is indicated by a line whose first five characters are "From ", but additional occurrences of "From " are not transformed to ">From " when storing messages because the extra message separator lines prevent mistaking such occurrences for the starts of subsequent messages.

Some :class:`Mailbox` methods implemented by :class:`MMDF` deserve special remarks:

```
.. method:: get_file(key)

    Using the file after calling :meth:`flush` or :meth:`close` on the
    :class:`MMDF` instance may yield unpredictable results or raise an
    exception.
```

```
.. method:: lock()
            unlock()

    Three locking mechanisms are used---dot locking and, if available, the
    :c:func:`flock` and :c:func:`lockf` system calls.
```

```
.. seealso::

    `mmdf man page from tin <http://www.tin.org/bin/man.cgi?section=5&topic=mmdf>`_
        A specification of MMDF format from the documentation of tin, a newsreader.

    `MMDF <https://en.wikipedia.org/wiki/MMDF>`_
        A Wikipedia article describing the Multichannel Memorandum Distribution
        Facility.
```

:class:`Message` objects

A subclass of the :mod:`email.message` module's :class:`~email.message.Message`. Subclasses of :class:`mailbox.Message` add mailbox-format-specific state and behavior.

If *message* is omitted, the new instance is created in a default, empty state. If *message* is an :class:`email.message.Message` instance, its contents are copied; furthermore, any format-specific information is converted insofar as possible if *message* is a :class:`Message` instance. If *message* is a string, a byte string, or a file, it should contain an RFC 2822-compliant message, which is read and parsed. Files should be open in binary mode, but text mode files are accepted for backward compatibility.

Unknown interpreted text role "class".

The format-specific state and behaviors offered by subclasses vary, but in general it is only the properties that are not specific to a particular mailbox that are supported (although presumably the properties are specific to a particular mailbox format). For example, file offsets for single-file mailbox formats and file names for directory-based mailbox formats are not retained, because they are only applicable to the original mailbox. But state such as whether a message has been read by the user or marked as important is retained, because it applies to the message itself.

There is no requirement that :class:`Message` instances be used to represent messages retrieved using :class:`Mailbox` instances. In some situations, the time and memory required to generate :class:`Message` representations might not be acceptable. For such situations, :class:`Mailbox` instances also offer string and file-like representations, and a custom message factory may be specified when a :class:`Mailbox` instance is initialized.

:class:`MaildirMessage`

A message with Maildir-specific behaviors. Parameter *message* has the same meaning as with the :class:`Message` constructor.

Typically, a mail user agent application moves all of the messages in the :file:`new` subdirectory to the :file:`cur` subdirectory after the first time the user opens and closes the mailbox, recording that the messages are old whether or not they've actually been read. Each message in :file:`cur` has an "info" section added to its file name to store information about its state. (Some mail readers may also add an "info" section to messages in :file:`new`.) The "info" section may take one of two forms: it may contain "2," followed by a list of standardized flags (e.g., "2,FR") or it may contain "1," followed by so-called experimental information. Standard flags for Maildir messages are as follows:

Flag	Meaning	Explanation
D	Draft	Under composition
F	Flagged	Marked as important
P	Passed	Forwarded, resent, or bounced
R	Replied	Replied to
S	Seen	Read
T	Trashed	Marked for subsequent deletion

:class:`MaildirMessage` instances offer the following methods:

```
.. method:: get_subdir()

   Return either "new" (if the message should be stored in the :file:`new`
   subdirectory) or "cur" (if the message should be stored in the :file:`cur`
   subdirectory).

   .. note::

      A message is typically moved from :file:`new` to :file:`cur` after its
      mailbox has been accessed, whether or not the message is has been
      read. A message ``msg`` has been read if ``"S" in msg.get_flags()`` is
      ``True``.
```

```
.. method:: set_subdir(subdir)

   Set the subdirectory the message should be stored in. Parameter *subdir*
   must be either "new" or "cur".
```

```
.. method:: get_flags()

   Return a string specifying the flags that are currently set. If the
   message complies with the standard Maildir format, the result is the
   concatenation in alphabetical order of zero or one occurrence of each of
   ``'D'``, ``'F'``, ``'P'``, ``'R'``, ``'S'``, and ``'T'``. The empty string
   is returned if no flags are set or if "info" contains experimental
   semantics.
```

```
.. method:: set_flags(flags)

   Set the flags specified by *flags* and unset all others.
```

```
.. method:: add_flag(flag)

   Set the flag(s) specified by *flag* without changing other flags. To add
   more than one flag at a time, *flag* may be a string of more than one
   character. The current "info" is overwritten whether or not it contains
   experimental information rather than flags.
```

```
.. method:: remove_flag(flag)

   Unset the flag(s) specified by *flag* without changing other flags. To
   remove more than one flag at a time, *flag* maybe a string of more than
   one character.  If "info" contains experimental information rather than
   flags, the current "info" is not modified.
```

```
.. method:: get_date()

   Return the delivery date of the message as a floating-point number
   representing seconds since the epoch.
```

When a :class:`MaildirMessage` instance is created based upon an :class:`mboxMessage` or :class:`MMDFMessage` instance, the :mailheader:`Status` and :mailheader:`X-Status` headers are omitted and the following conversions take place:

Resulting state	:class:`mboxMessage` or :class:`MMDFMessage` state
"cur" subdirectory	O flag
F flag	F flag
R flag	A flag
S flag	R flag
T flag	D flag

When a :class:`MaildirMessage` instance is created based upon an :class:`MHMessage` instance, the following conversions take place:

Resulting state	**:class:`MHMessage` state**
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 931);** *backlink* **Unknown interpreted text role "class".**
"cur" subdirectory	"unseen" sequence
"cur" subdirectory and S flag	no "unseen" sequence
F flag	"flagged" sequence
R flag	"replied" sequence

When a :class:`MaildirMessage` instance is created based upon a :class:`BabylMessage` instance, the following conversions take place:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 941);** *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 941);** *backlink*
>
> Unknown interpreted text role "class".

Resulting state	**:class:`BabylMessage` state**
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 946);** *backlink* **Unknown interpreted text role "class".**
"cur" subdirectory	"unseen" label
"cur" subdirectory and S flag	no "unseen" label
P flag	"forwarded" or "resent" label
R flag	"answered" label
T flag	"deleted" label

:class:`mboxMessage`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 961);** *backlink*
>
> Unknown interpreted text role "class".

A message with mbox-specific behaviors. Parameter *message* has the same meaning as with the :class:`Message` constructor.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 967);** *backlink*
>
> Unknown interpreted text role "class".

Messages in an mbox mailbox are stored together in a single file. The sender's envelope address and the time of delivery are typically stored in a line beginning with "From " that is used to indicate the start of a message, though there is considerable variation in the exact format of this data among mbox implementations. Flags that indicate the state of the message, such as whether it has been read or marked as important, are typically stored in :mailheader:`Status` and :mailheader:`X-Status` headers.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 970);** *backlink*
>
> Unknown interpreted text role "mailheader".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 970);** *backlink*
>
> Unknown interpreted text role "mailheader".

Conventional flags for mbox messages are as follows:

Flag	Meaning	Explanation
R	Read	Read
O	Old	Previously detected by MUA
D	Deleted	Marked for subsequent deletion
F	Flagged	Marked as important
A	Answered	Replied to

The "R" and "O" flags are stored in the :mailheader:`Status` header, and the "D", "F", and "A" flags are stored in the :mailheader:`X-Status` header. The flags and headers typically appear in the order mentioned.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 994);** *backlink*
>
> Unknown interpreted text role "mailheader".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, **line 994);** *backlink*

Unknown interpreted text role "mailheader".

:class:`mboxMessage` instances offer the following methods:

When an :class:`mboxMessage` instance is created based upon a :class:`MaildirMessage` instance, a "From " line is generated based upon the :class:`MaildirMessage` instance's delivery date, and the following conversions take place:

Resulting state	:class:`MaildirMessage` state
R flag	S flag
O flag	"cur" subdirectory
D flag	T flag
F flag	F flag
A flag	R flag

When an :class:`mboxMessage` instance is created based upon an :class:`MHMessage` instance, the following conversions take place:

Resulting state	:class:`MHMessage` state
R flag and O flag	no "unseen" sequence
O flag	"unseen" sequence
F flag	"flagged" sequence
A flag	"replied" sequence

When an :class:`mboxMessage` instance is created based upon a :class:`BabylMessage` instance, the following conversions take place:

Resulting state	:class:`BabylMessage` state
R flag and O flag	no "unseen" label
O flag	"unseen" label
D flag	"deleted" label
A flag	"answered" label

When a :class:`Message` instance is created based upon an :class:`MMDFMessage` instance, the "From " line is copied and all flags directly correspond:

Resulting state	:class:`MMDFMessage` state
	System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1100); *backlink* **Unknown interpreted text role "class".**
R flag	R flag
O flag	O flag
D flag	D flag
F flag	F flag
A flag	A flag

:class:`MHMessage`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1115); *backlink*

Unknown interpreted text role "class".

A message with MH-specific behaviors. Parameter *message* has the same meaning as with the :class:`Message` constructor.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1121); *backlink*

Unknown interpreted text role "class".

MH messages do not support marks or flags in the traditional sense, but they do support sequences, which are logical groupings of arbitrary messages. Some mail reading programs (although not the standard :program:`mh` and :program:`nmh`) use sequences in much the same way flags are used with other formats, as follows:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1124); *backlink*

Unknown interpreted text role "program".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1124); *backlink*

Unknown interpreted text role "program".

Sequence	Explanation
unseen	Not read, but previously detected by MUA
replied	Replied to
flagged	Marked as important

:class:`MHMessage` instances offer the following methods:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1140); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1143)

Unknown directive type "method".

```
.. method:: get_sequences()

   Return a list of the names of sequences that include this message.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1148)

Unknown directive type "method".

```
.. method:: set_sequences(sequences)

   Set the list of sequences that include this message.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1153)

Unknown directive type "method".

```
.. method:: add_sequence(sequence)

   Add *sequence* to the list of sequences that include this message.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst`, line 1158)

Unknown directive type "method".

```
.. method:: remove_sequence(sequence)

   Remove *sequence* from the list of sequences that include this message.
```

When an :class:`MHMessage` instance is created based upon a :class:`MaildirMessage` instance, the following conversions take place:

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1162);** *backlink*

Unknown interpreted text role "class".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1162);** *backlink*

Unknown interpreted text role "class".

</div>

Resulting state	**:class:`MaildirMessage` state** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1167);** *backlink* **Unknown interpreted text role "class".**
"unseen" sequence	no S flag
"replied" sequence	R flag
"flagged" sequence	F flag

When an :class:`MHMessage` instance is created based upon an :class:`mboxMessage` or :class:`MMDFMessage` instance, the :mailheader:`Status` and :mailheader:`X-Status` headers are omitted and the following conversions take place:

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1175);** *backlink*

Unknown interpreted text role "class".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1175);** *backlink*

Unknown interpreted text role "class".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1175);** *backlink*

Unknown interpreted text role "class".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1175);** *backlink*

Unknown interpreted text role "mailheader".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1175);** *backlink*

Unknown interpreted text role "mailheader".

</div>

Resulting state	**:class:`mboxMessage` or :class:`MMDFMessage` state** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1182);** *backlink* **Unknown interpreted text role "class".** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1182);** *backlink* **Unknown interpreted text role "class".**
"unseen" sequence	no R flag
"replied" sequence	A flag
"flagged" sequence	F flag

When an :class:`MHMessage` instance is created based upon a :class:`BabylMessage` instance, the following conversions take place:

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1191);** *backlink*

Unknown interpreted text role "class".

</div>

<div style="border:1px solid black; padding:8px">

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1191);** *backlink*

Unknown interpreted text role "class".

</div>

	:class:`BabylMessage` state
Resulting state	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1196`); *backlink* **Unknown interpreted text role "class".**
"unseen" sequence	"unseen" label
"replied" sequence	"answered" label

:class:`BabylMessage`

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1205`); *backlink*
>
> Unknown interpreted text role "class".

A message with Babyl-specific behaviors. Parameter *message* has the same meaning as with the :class:`Message` constructor.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1211`); *backlink*
>
> Unknown interpreted text role "class".

Certain message labels, called :dfn:`attributes`, are defined by convention to have special meanings. The attributes are as follows:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1214`); *backlink*
>
> Unknown interpreted text role "dfn".

Label	Explanation
unseen	Not read, but previously detected by MUA
deleted	Marked for subsequent deletion
filed	Copied to another file or mailbox
answered	Replied to
forwarded	Forwarded
edited	Modified by the user
resent	Resent

By default, Rmail displays only visible headers. The :class:`BabylMessage` class, though, uses the original headers because they are more complete. Visible headers may be accessed explicitly if desired.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1235`); *backlink*
>
> Unknown interpreted text role "class".

:class:`BabylMessage` instances offer the following methods:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1239`); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1242`)
>
> Unknown directive type "method".
>
> ```
> .. method:: get_labels()
>
>     Return a list of labels on the message.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1247`)
>
> Unknown directive type "method".
>
> ```
> .. method:: set_labels(labels)
>
>     Set the list of labels on the message to *labels*.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1252`)
>
> Unknown directive type "method".
>
> ```
> .. method:: add_label(label)
>
>     Add *label* to the list of labels on the message.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1257`)
>
> Unknown directive type "method".
>
> ```
> .. method:: remove_label(label)
> ```

```
     Remove *label* from the list of labels on the message.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1262**)

Unknown directive type "method".

```
   .. method:: get_visible()

       Return an :class:`Message` instance whose headers are the message's
       visible headers and whose body is empty.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1268**)

Unknown directive type "method".

```
   .. method:: set_visible(visible)

       Set the message's visible headers to be the same as the headers in
       *message*.  Parameter *visible* should be a :class:`Message` instance, an
       :class:`email.message.Message` instance, a string, or a file-like object
       (which should be open in text mode).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1276**)

Unknown directive type "method".

```
   .. method:: update_visible()

       When a :class:`BabylMessage` instance's original headers are modified, the
       visible headers are not automatically modified to correspond. This method
       updates the visible headers as follows: each visible header with a
       corresponding original header is set to the value of the original header,
       each visible header without a corresponding original header is removed,
       and any of :mailheader:`Date`, :mailheader:`From`, :mailheader:`Reply-To`,
       :mailheader:`To`, :mailheader:`CC`, and :mailheader:`Subject` that are
       present in the original headers but not the visible headers are added to
       the visible headers.
```

When a :class:`BabylMessage` instance is created based upon a :class:`MaildirMessage` instance, the following conversions take place:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1288**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1288**); *backlink*

Unknown interpreted text role "class".

	:class:`MaildirMessage` state
Resulting state	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1293`); *backlink* **Unknown interpreted text role "class".**
"unseen" label	no S flag
"deleted" label	T flag
"answered" label	R flag
"forwarded" label	P flag

When a :class:`BabylMessage` instance is created based upon an :class:`mboxMessage` or :class:`MMDFMessage` instance, the :mailheader:`Status` and :mailheader:`X-Status` headers are omitted and the following conversions take place:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1303**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1303**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1303**); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst`, **line 1303**); *backlink*

Unknown interpreted text role "mailheader".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1303`); *backlink*

Unknown interpreted text role "mailheader".

Resulting state	**:class:`mboxMessage`** or **:class:`MMDFMessage`** state **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1310`); *backlink* **Unknown interpreted text role "class".** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1310`); *backlink* **Unknown interpreted text role "class".**
"unseen" label	no R flag
"deleted" label	D flag
"answered" label	A flag

When a :class:`BabylMessage` instance is created based upon an :class:`MHMessage` instance, the following conversions take place:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1319`); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1319`); *backlink*

Unknown interpreted text role "class".

Resulting state	**:class:`MHMessage`** state **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1324`); *backlink* **Unknown interpreted text role "class".**
"unseen" label	"unseen" sequence
"answered" label	"replied" sequence

:class:`MMDFMessage`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1333`); *backlink*

Unknown interpreted text role "class".

A message with MMDF-specific behaviors. Parameter *message* has the same meaning as with the :class:`Message` constructor.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1339`); *backlink*

Unknown interpreted text role "class".

As with message in an mbox mailbox, MMDF messages are stored with the sender's address and the delivery date in an initial line beginning with "From ". Likewise, flags that indicate the state of the message are typically stored in :mailheader:`Status` and :mailheader:`X-Status` headers.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1342`); *backlink*

Unknown interpreted text role "mailheader".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1342`); *backlink*

Unknown interpreted text role "mailheader".

Conventional flags for MMDF messages are identical to those of mbox message and are as follows:

Flag	Meaning	Explanation
R	Read	Read
O	Old	Previously detected by MUA
D	Deleted	Marked for subsequent deletion
F	Flagged	Marked as important
A	Answered	Replied to

The "R" and "O" flags are stored in the :mailheader:`Status` header, and the "D", "F", and "A" flags are stored in the :mailheader:`X-Status` header. The flags and headers typically appear in the order mentioned.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)mailbox.rst, line 1364`); *backlink*

Unknown interpreted text role "mailheader".

:class:`MMDFMessage` instances offer the following methods, which are identical to those offered by :class:`mboxMessage`:

```
.. method:: get_from()

   Return a string representing the "From " line that marks the start of the
   message in an mbox mailbox. The leading "From " and the trailing newline
   are excluded.
```

```
.. method:: set_from(from_, time_=None)

   Set the "From " line to *from_*, which should be specified without a
   leading "From " or trailing newline. For convenience, *time_* may be
   specified and will be formatted appropriately and appended to *from_*. If
   *time_* is specified, it should be a :class:`time.struct_time` instance, a
   tuple suitable for passing to :meth:`time.strftime`, or ``True`` (to use
   :meth:`time.gmtime`).
```

```
.. method:: get_flags()

   Return a string specifying the flags that are currently set. If the message
   complies with the conventional format, the result is the
   concatenation in the following order of zero or one occurrence of each of
   ``'R'``, ``'O'``, ``'D'``, ``'F'``, and ``'A'``.
```

```
.. method:: set_flags(flags)

   Set the flags specified by *flags* and unset all others. Parameter *flags*
   should be the concatenation in any order of zero or more occurrences of
   each of ``'R'``, ``'O'``, ``'D'``, ``'F'``, and ``'A'``.
```

```
.. method:: add_flag(flag)

   Set the flag(s) specified by *flag* without changing other flags. To add
   more than one flag at a time, *flag* may be a string of more than one
   character.
```

```
.. method:: remove_flag(flag)

   Unset the flag(s) specified by *flag* without changing other flags. To
   remove more than one flag at a time, *flag* maybe a string of more than
   one character.
```

When an :class:`MMDFMessage` instance is created based upon a :class:`MaildirMessage` instance, a "From " line is generated based upon the :class:`MaildirMessage` instance's delivery date, and the following conversions take place:

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1417); backlink`

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1417); backlink`

Unknown interpreted text role "class".

Resulting state	**:class:`MaildirMessage` state** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1424); backlink` **Unknown interpreted text role "class".**
R flag	S flag
O flag	"cur" subdirectory
D flag	T flag
F flag	F flag
A flag	R flag

When an :class:`MMDFMessage` instance is created based upon an :class:`MHMessage` instance, the following conversions take place:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1436); backlink`

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1436); backlink`

Unknown interpreted text role "class".

Resulting state	**:class:`MHMessage` state** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1441); backlink` **Unknown interpreted text role "class".**
R flag and O flag	no "unseen" sequence
O flag	"unseen" sequence
F flag	"flagged" sequence
A flag	"replied" sequence

When an :class:`MMDFMessage` instance is created based upon a :class:`BabylMessage` instance, the following conversions take place:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1451); backlink`

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1451); backlink`

Unknown interpreted text role "class".

Resulting state	**:class:`BabylMessage` state** **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1456); backlink` **Unknown interpreted text role "class".**
R flag and O flag	no "unseen" label
O flag	"unseen" label
D flag	"deleted" label
A flag	"answered" label

When an :class:`MMDFMessage` instance is created based upon an :class:`mboxMessage` instance, the "From " line is copied and all flags directly correspond:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1466); backlink`

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1466); backlink`

Unknown interpreted text role "class".

	:class:`mboxMessage` state
Resulting state	**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1472`); *backlink* **Unknown interpreted text role "class".**
R flag	R flag
O flag	O flag
D flag	D flag
F flag	F flag
A flag	A flag

Exceptions

The following exception classes are defined in the :mod:`mailbox` module:

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1488`); *backlink*
>
> Unknown interpreted text role "mod".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1491`)
>
> Unknown directive type "exception".
>
> ```
>    .. exception:: Error()
>
>        The based class for all other module-specific exceptions.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1496`)
>
> Unknown directive type "exception".
>
> ```
>    .. exception:: NoSuchMailboxError()
>
>        Raised when a mailbox is expected but is not found, such as when instantiating a
>        :class:`Mailbox` subclass with a path that does not exist (and with the *create*
>        parameter set to ``False``), or when opening a folder that does not exist.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1503`)
>
> Unknown directive type "exception".
>
> ```
>    .. exception:: NotEmptyError()
>
>        Raised when a mailbox is not empty but is expected to be, such as when deleting
>        a folder that contains messages.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1509`)
>
> Unknown directive type "exception".
>
> ```
>    .. exception:: ExternalClashError()
>
>        Raised when some mailbox-related condition beyond the control of the program
>        causes it to be unable to proceed, such as when failing to acquire a lock that
>        another program already holds a lock, or when a uniquely-generated file name
>        already exists.
> ```

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)mailbox.rst, line 1517`)
>
> Unknown directive type "exception".
>
> ```
>    .. exception:: FormatError()
>
>        Raised when the data in a file cannot be parsed, such as when an :class:`MH`
>        instance attempts to read a corrupted :file:`.mh_sequences` file.
> ```

Examples

A simple example of printing the subjects of all messages in a mailbox that seem interesting:

```
import mailbox
for message in mailbox.mbox('~/mbox'):
    subject = message['subject']      # Could possibly be None.
    if subject and 'python' in subject.lower():
        print(subject)
```

To copy all mail from a Babyl mailbox to an MH mailbox, converting all of the format-specific information that can be converted:

```
import mailbox
destination = mailbox.MH('~/Mail')
destination.lock()
for message in mailbox.Babyl('~/RMAIL'):
```

```
        destination.add(mailbox.MHMessage(message))
destination.flush()
destination.unlock()
```

This example sorts mail from several mailing lists into different mailboxes, being careful to avoid mail corruption due to concurrent modification by other programs, mail loss due to interruption of the program, or premature termination due to malformed messages in the mailbox:

```
import mailbox
import email.errors

list_names = ('python-list', 'python-dev', 'python-bugs')

boxes = {name: mailbox.mbox('~/email/%s' % name) for name in list_names}
inbox = mailbox.Maildir('~/Maildir', factory=None)

for key in inbox.iterkeys():
    try:
        message = inbox[key]
    except email.errors.MessageParseError:
        continue                 # The message is malformed. Just leave it.

    for name in list_names:
        list_id = message['list-id']
        if list_id and name in list_id:
            # Get mailbox to use
            box = boxes[name]

            # Write copy to disk before removing original.
            # If there's a crash, you might duplicate a message, but
            # that's better than losing a message completely.
            box.lock()
            box.add(message)
            box.flush()
            box.unlock()

            # Remove original message
            inbox.lock()
            inbox.discard(key)
            inbox.flush()
            inbox.unlock()
            break                # Found destination, so stop looking.

for box in boxes.itervalues():
    box.close()
```