

---

# **mailib Documentation**

*Release 0.1.3*

**Joseph Kocherhans**

**Oct 31, 2017**



---

## Contents

---

<b>1</b>	<b>Messages</b>	<b>3</b>
1.1	Creating Message objects . . . . .	3
1.2	Message attributes . . . . .	3
<b>2</b>	<b>Changes in maillib</b>	<b>5</b>
2.1	Release 0.1.4 (In development) . . . . .	5
2.2	Release 0.1.3 (2009-11-23) . . . . .	5
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



Contents:



maillib's `Message` class provides a simple wrapper around Python's `email.message.Message` that tries to decode values to unicode automatically, but leaves access to the underlying object in case you need to do something more complex.

### Creating Message objects

**classmethod** `Message.from_string` (*string*)

Creates and returns a `maillib.Message` from a string that contains a raw mime message.

**classmethod** `Message.from_file` (*file*)

Creates and returns a `maillib.Message` from a file-like object that contains a raw mime message.

**classmethod** `Message.from_message` (*message*)

Creates and returns a `maillib.Message` from an `email.message.Message` instance.

### Message attributes

**class** `maillib.Message`

**subject**

The value of the message's *Subject* header.

**sender**

A (name, email\_address) tuple from the message's *From* header. For example: ('John Doe', 'john@example.com')

**to**

A list of (name, email\_address) tuples for every recipient in the message's *To* header. For example: [('John Doe', 'john@example.com'), ('Jane Doe', 'jane@example.com')]

**cc**

A list of `(name, email_address)` tuples for every recipient in the message's `Cc` header. For example: `[('John Doe', 'john@example.com'), ('Jane Doe', 'jane@example.com')]`

**date**

A `datetime.datetime` object representing the message's `Date` header.

<b>Warning:</b> The value is always UTC, so you might want to convert it to local time.
---

**body**

The plain text body of the message if one exists, otherwise `None`.

In practice, we do a depth-first traversal of the message using `message.walk()` and return the first payload that has a `Content-type` of `text/plain`.

**html**

The HTML body of the message if one exists, otherwise `None`.

In practice, we do a depth-first traversal of the message using `message.walk()` and return the first payload that has a `Content-type` of `text/html`.

**attachments()**

Returns an iterator over `(filename, content)` tuples for every attachment in the message.

**references**

A list of everything that looks like a `Message-id` in the `References` header, and the first thing that looks like a `Message-id` in the `In-reply-to` header. See [jwz's threading article](http://www.jwz.org/doc/threading.html)<sup>1</sup> for why you'd want to do this.

**headers**

A dictionary like interface to decoded message headers.

**raw\_headers**

A dictionary like interface to raw message headers.

---

<sup>1</sup> <http://www.jwz.org/doc/threading.html>



#### **Release 0.1.4 (In development)**

- Fixed a bug with classmethod constructors that were always returning message instances rather than instances of the subclass.
- Added a utils module with the method for extracting list ids from headers.
- Moved normalize\_subject to maillib.utils.

#### **Release 0.1.3 (2009-11-23)**

- First public release.



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**m**

maillib, 1



## A

attachments() (maillib.Message method), 4

## B

body (maillib.Message attribute), 4

## C

cc (maillib.Message attribute), 3

## D

date (maillib.Message attribute), 4

## F

from\_file() (maillib.Message class method), 3

from\_message() (maillib.Message class method), 3

from\_string() (maillib.Message class method), 3

## H

headers (maillib.Message attribute), 4

html (maillib.Message attribute), 4

## M

maillib (module), 1

Message (class in maillib), 3

## R

raw\_headers (maillib.Message attribute), 4

references (maillib.Message attribute), 4

## S

sender (maillib.Message attribute), 3

subject (maillib.Message attribute), 3

## T

to (maillib.Message attribute), 3