

## :mod:`email.message`: Representing an email message

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.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]email.message.rst, line 4)

Unknown directive type "module".

```
.. module:: email.message
   :synopsis: The base class representing email messages.
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 6)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: R. David Murray <rdmurray@bitdance.com>
```

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 7)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: R. David Murray <rdmurray@bitdance.com>,
                  Barry A. Warsaw <barry@python.org>
```

Source code: :source:`Lib/email/message.py`

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 10); [backlink](#)

Unknown interpreted text role "source".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 14)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6 [1]
```

The central class in the :mod:`email` package is the :class:`EmailMessage` class, imported from the :mod:`email.message` module. It is the base class for the :mod:`email` object model. :class:`EmailMessage` provides the core functionality for setting and querying header fields, for accessing message bodies, and for creating or modifying structured messages.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 16); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 16); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 16); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 16); [backlink](#)**

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 16); [backlink](#)**

Unknown interpreted text role "class".

An email message consists of *headers* and a *payload* (which is also referred to as the *content*). Headers are [RFC 5322](#) or [RFC 6532](#) style field names and values, where the field name and value are separated by a colon. The colon is not part of either the field name or the field value. The payload may be a simple text message, or a binary object, or a structured sequence of sub-messages each with their own set of headers and their own payload. The latter type of payload is indicated by the message having a MIME type such as `:mimetype:'multipart/*'` or `:mimetype:'message/rfc822'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 22); [backlink](#)**

Unknown interpreted text role "mimetype".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 22); [backlink](#)**

Unknown interpreted text role "mimetype".

The conceptual model provided by an `:class:'EmailMessage'` object is that of an ordered dictionary of headers coupled with a *payload* that represents the [RFC 5322](#) body of the message, which might be a list of sub-`EmailMessage` objects. In addition to the normal dictionary methods for accessing the header names and values, there are methods for accessing specialized information from the headers (for example the MIME content type), for operating on the payload, for generating a serialized version of the message, and for recursively walking over the object tree.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 31); [backlink](#)**

Unknown interpreted text role "class".

The `:class:'EmailMessage'` dictionary-like interface is indexed by the header names, which must be ASCII values. The values of the dictionary are strings with some extra methods. Headers are stored and returned in case-preserving form, but field names are matched case-insensitively. Unlike a real dict, there is an ordering to the keys, and there can be duplicate keys. Additional methods are provided for working with headers that have duplicate keys.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 40); [backlink](#)**

Unknown interpreted text role "class".

The *payload* is either a string or bytes object, in the case of simple message objects, or a list of `:class:'EmailMessage'` objects, for MIME container documents such as `:mimetype:'multipart/*'` and `:mimetype:'message/rfc822'` message objects.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.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]email.message.rst, line 47); [backlink](#)**

Unknown interpreted text role "mimetype".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 47); [backlink](#)**

Unknown interpreted text role "mimetype".

If *policy* is specified use the rules it specifies to update and serialize the representation of the message. If *policy* is not set, use the `:class:`~email.policy.default`` policy, which follows the rules of the email RFCs except for line endings (instead of the RFC mandated `\r\n`, it uses the Python standard `\n` line endings). For more information see the `:mod:`~email.policy`` documentation.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 55); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 55); [backlink](#)**

Unknown interpreted text role "mod".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 62)**

Unknown directive type "method".

```
.. method:: as_string(unixfrom=False, maxheaderlen=None, policy=None)
```

Return the entire message flattened as a string. When optional *\*unixfrom\** is true, the envelope header is included in the returned string. *\*unixfrom\** defaults to `False`. For backward compatibility with the base `:class:`~email.message.Message`` class *\*maxheaderlen\** is accepted, but defaults to `None`, which means that by default the line length is controlled by the `:attr:`~email.policy.EmailPolicy.max_line_length`` of the policy. The *\*policy\** argument may be used to override the default policy obtained from the message instance. This can be used to control some of the formatting produced by the method, since the specified *\*policy\** will be passed to the `:class:`~email.generator.Generator``.

Flattening the message may trigger changes to the `:class:`~EmailMessage`` if defaults need to be filled in to complete the transformation to a string (for example, MIME boundaries may be generated or modified).

Note that this method is provided as a convenience and may not be the most useful way to serialize messages in your application, especially if you are dealing with multiple messages. See `:class:`~email.generator.Generator`` for a more flexible API for serializing messages. Note also that this method is restricted to producing messages serialized as "7 bit clean" when `:attr:`~email.policy.EmailPolicy.utf8`` is `False`, which is the default.

```
.. versionchanged:: 3.6 the default behavior when *maxheaderlen*
   is not specified was changed from defaulting to 0 to defaulting
   to the value of *max_line_length* from the policy.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 93)**

Unknown directive type "method".

```
.. method:: __str__()
```

Equivalent to `as_string(policy=self.policy.clone(utf8=True))`. Allows `str(msg)` to produce a string containing the serialized message in a readable format.

```
.. versionchanged:: 3.4 the method was changed to use `utf8=True`,
   thus producing an :rfc:6531-like message representation, instead of
   being a direct alias for :meth:`as_string`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 104)**

Unknown directive type "method".

```
.. method:: as_bytes(unixfrom=False, policy=None)
```

Return the entire message flattened as a bytes object. When optional `*unixfrom*` is true, the envelope header is included in the returned string. `*unixfrom*` defaults to `False`. The `*policy*` argument may be used to override the default policy obtained from the message instance. This can be used to control some of the formatting produced by the method, since the specified `*policy*` will be passed to the `:class:`~email.generator.BytesGenerator``.

Flattening the message may trigger changes to the `:class:`~EmailMessage`` if defaults need to be filled in to complete the transformation to a string (for example, MIME boundaries may be generated or modified).

Note that this method is provided as a convenience and may not be the most useful way to serialize messages in your application, especially if you are dealing with multiple messages. See `:class:`~email.generator.BytesGenerator`` for a more flexible API for serializing messages.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 125)**

Unknown directive type "method".

```
.. method:: __bytes__()
```

Equivalent to `:meth:`~as_bytes``. Allows `bytes(msg)` to produce a bytes object containing the serialized message.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 131)**

Unknown directive type "method".

```
.. method:: is_multipart()
```

Return `True` if the message's payload is a list of sub-`:class:`~EmailMessage`` objects, otherwise return `False`. When `:meth:`~is_multipart`` returns `False`, the payload should be a string object (which might be a CTE encoded binary payload). Note that `:meth:`~is_multipart`` returning `True` does not necessarily mean that `"msg.get_content_maintype() == 'multipart'"` will return the `True`. For example, `is_multipart` will return `True` when the `:class:`~EmailMessage`` is of type `message/rfc822`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 143)**

Unknown directive type "method".

```
.. method:: set_unixfrom(unixfrom)
```

Set the message's envelope header to `*unixfrom*`, which should be a string. (See `:class:`~mailbox.mboxMessage`` for a brief description of this header.)

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 150)**

Unknown directive type "method".

```
.. method:: get_unixfrom()
```

Return the message's envelope header. Defaults to `None` if the envelope header was never set.

The following methods implement the mapping-like interface for accessing the message's headers. Note that there are some semantic

differences between these methods and a normal mapping (i.e. dictionary) interface. For example, in a dictionary there are no duplicate keys, but here there may be duplicate message headers. Also, in dictionaries there is no guaranteed order to the keys returned by `meth:'keys'`, but in an `:class:'EmailMessage'` object, headers are always returned in the order they appeared in the original message, or in which they were added to the message later. Any header deleted and then re-added is always appended to the end of the header list.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 156); [backlink](#)**

Unknown interpreted text role "meth".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 156); [backlink](#)**

Unknown interpreted text role "class".

These semantic differences are intentional and are biased toward convenience in the most common use cases.

Note that in all cases, any envelope header present in the message is not included in the mapping interface.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 174)**

Unknown directive type "method".

```
.. method:: __len__()
```

Return the total number of headers, including duplicates.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 179)**

Unknown directive type "method".

```
.. method:: __contains__(name)
```

Return ``True`` if the message object has a field named \*name\*. Matching is done without regard to case and \*name\* does not include the trailing colon. Used for the ``in`` operator. For example::

```
if 'message-id' in myMessage:
    print('Message-ID:', myMessage['message-id'])
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 189)**

Unknown directive type "method".

```
.. method:: __getitem__(name)
```

Return the value of the named header field. \*name\* does not include the colon field separator. If the header is missing, ``None`` is returned; a `:exc:`KeyError`` is never raised.

Note that if the named field appears more than once in the message's headers, exactly which of those field values will be returned is undefined. Use the `:meth:`get_all`` method to get the values of all the extant headers named \*name\*.

Using the standard (non-``compat32``) policies, the returned value is an instance of a subclass of `:class:`email.headerregistry.BaseHeader``.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 204)**

Unknown directive type "method".

```
.. method:: __setitem__(name, val)
```

Add a header to the message with field name `*name*` and value `*val*`. The field is appended to the end of the message's existing headers.

Note that this does *not* overwrite or delete any existing header with the same name. If you want to ensure that the new header is the only one present in the message with field name `*name*`, delete the field first, e.g.::

```
del msg['subject']
msg['subject'] = 'Python roolz!'
```

If the `:mod:`policy`` defines certain headers to be unique (as the standard policies do), this method may raise a `:exc:`ValueError`` when an attempt is made to assign a value to such a header when one already exists. This behavior is intentional for consistency's sake, but do not depend on it as we may choose to make such assignments do an automatic deletion of the existing header in the future.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 224)**

Unknown directive type "method".

```
.. method:: __delitem__(name)
```

Delete all occurrences of the field with name `*name*` from the message's headers. No exception is raised if the named field isn't present in the headers.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 231)**

Unknown directive type "method".

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

Return a list of all the message's header field names.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 236)**

Unknown directive type "method".

```
.. method:: values()
```

Return a list of all the message's field values.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 241)**

Unknown directive type "method".

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

Return a list of 2-tuples containing all the message's field headers and values.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 247)**

Unknown directive type "method".

```
.. method:: get(name, failobj=None)
```

Return the value of the named header field. This is identical to `:meth:`__getitem__`` except that optional `*failobj*` is returned if the

named header is missing (\*failobj\* defaults to ``None``).

Here are some additional useful header related methods:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 257)**

Unknown directive type "method".

```
.. method:: get_all(name, failobj=None)
```

Return a list of all the values for the field named \*name\*. If there are no such named headers in the message, \*failobj\* is returned (defaults to ``None``).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 264)**

Unknown directive type "method".

```
.. method:: add_header(_name, _value, **_params)
```

Extended header setting. This method is similar to :meth:`\_\_setitem\_\_` except that additional header parameters can be provided as keyword arguments. \*\_name\* is the header field to add and \*\_value\* is the \*primary\* value for the header.

For each item in the keyword argument dictionary \*\_params\*, the key is taken as the parameter name, with underscores converted to dashes (since dashes are illegal in Python identifiers). Normally, the parameter will be added as ``key="value"`` unless the value is ``None``, in which case only the key will be added.

If the value contains non-ASCII characters, the charset and language may be explicitly controlled by specifying the value as a three tuple in the format ``(CHARSET, LANGUAGE, VALUE)`, where ``CHARSET`` is a string naming the charset to be used to encode the value, ``LANGUAGE`` can usually be set to ``None`` or the empty string (see :rfc:`2231` for other possibilities), and ``VALUE`` is the string value containing non-ASCII code points. If a three tuple is not passed and the value contains non-ASCII characters, it is automatically encoded in :rfc:`2231` format using a ``CHARSET`` of ``utf-8`` and a ``LANGUAGE`` of ``None``.

Here is an example::

```
msg.add_header('Content-Disposition', 'attachment', filename='bud.gif')
```

This will add a header that looks like ::

```
Content-Disposition: attachment; filename="bud.gif"
```

An example of the extended interface with non-ASCII characters::

```
msg.add_header('Content-Disposition', 'attachment',
               filename=('iso-8859-1', '', 'FuÃballer.ppt'))
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 301)**

Unknown directive type "method".

```
.. method:: replace_header(_name, _value)
```

Replace a header. Replace the first header found in the message that matches \*\_name\*, retaining header order and field name case of the original header. If no matching header is found, raise a :exc:`KeyError`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 309)

Unknown directive type "method".

```
.. method:: get_content_type()
```

Return the message's content type, coerced to lower case of the form :mimetype: maintype/subtype`. If there is no :mailheader: `Content-Type` header in the message return the value returned by :meth:`get\_default\_type`. If the :mailheader: `Content-Type` header is invalid, return ``text/plain``.

(According to :rfc:`2045`, messages always have a default type, :meth:`get\_content\_type` will always return a value. :rfc:`2045` defines a message's default type to be :mimetype: `text/plain` unless it appears inside a :mimetype: `multipart/digest` container, in which case it would be :mimetype: `message/rfc822`. If the :mailheader: `Content-Type` header has an invalid type specification, :rfc:`2045` mandates that the default type be :mimetype: `text/plain`.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 326)

Unknown directive type "method".

```
.. method:: get_content_maintype()
```

Return the message's main content type. This is the :mimetype: `maintype` part of the string returned by :meth:`get\_content\_type`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 332)

Unknown directive type "method".

```
.. method:: get_content_subtype()
```

Return the message's sub-content type. This is the :mimetype: `subtype` part of the string returned by :meth:`get\_content\_type`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 338)

Unknown directive type "method".

```
.. method:: get_default_type()
```

Return the default content type. Most messages have a default content type of :mimetype: `text/plain`, except for messages that are subparts of :mimetype: `multipart/digest` containers. Such subparts have a default content type of :mimetype: `message/rfc822`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 346)

Unknown directive type "method".

```
.. method:: set_default_type ctype)
```

Set the default content type. \*ctype\* should either be :mimetype: `text/plain` or :mimetype: `message/rfc822`, although this is not enforced. The default content type is not stored in the :mailheader: `Content-Type` header, so it only affects the return value of the ``get\_content\_type`` methods when no :mailheader: `Content-Type` header is present in the message.



**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 356)**

Unknown directive type "method".

```
.. method:: set_param(param, value, header='Content-Type', requote=True, \
                      charset=None, language='', replace=False)
```

Set a parameter in the `:mailheader:Content-Type` header. If the parameter already exists in the header, replace its value with `*value*`. When `*header*` is `Content-Type` (the default) and the header does not yet exist in the message, add it, set its value to `:mimetype:text/plain`, and append the new parameter value. Optional `*header*` specifies an alternative header to `:mailheader:Content-Type`.

If the value contains non-ASCII characters, the `charset` and `language` may be explicitly specified using the optional `*charset*` and `*language*` parameters. Optional `*language*` specifies the `:rfc:2231` language, defaulting to the empty string. Both `*charset*` and `*language*` should be strings. The default is to use the `utf8` `*charset*` and `None` for the `*language*`.

If `*replace*` is `False` (the default) the header is moved to the end of the list of headers. If `*replace*` is `True`, the header will be updated in place.

Use of the `*requote*` parameter with `:class:EmailMessage` objects is deprecated.

Note that existing parameter values of headers may be accessed through the `:attr:~email.headerregistry.BaseHeader.params` attribute of the header value (for example, `msg['Content-Type'].params['charset']`).

```
.. versionchanged:: 3.4 ``replace`` keyword was added.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 387)**

Unknown directive type "method".

```
.. method:: del_param(param, header='content-type', requote=True)
```

Remove the given parameter completely from the `:mailheader:Content-Type` header. The header will be re-written in place without the parameter or its value. Optional `*header*` specifies an alternative to `:mailheader:Content-Type`.

Use of the `*requote*` parameter with `:class:EmailMessage` objects is deprecated.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 398)**

Unknown directive type "method".

```
.. method:: get_filename(failobj=None)
```

Return the value of the `filename` parameter of the `:mailheader:Content-Disposition` header of the message. If the header does not have a `filename` parameter, this method falls back to looking for the `name` parameter on the `:mailheader:Content-Type` header. If neither is found, or the header is missing, then `*failobj*` is returned. The returned string will always be unquoted as per `:func:email.utils.unquote`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]email.message.rst, line 409)**

Unknown directive type "method".

```
.. method:: get_boundary(failobj=None)
```

Return the value of the ``boundary`` parameter of the :mailheader:`Content-Type` header of the message, or \*failobj\* if either the header is missing, or has no ``boundary`` parameter. The returned string will always be unquoted as per :func:`email.utils.unquote`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 417)**

Unknown directive type "method".

```
.. method:: set_boundary(boundary)
```

Set the ``boundary`` parameter of the :mailheader:`Content-Type` header to \*boundary\*. :meth:`set\_boundary` will always quote \*boundary\* if necessary. A :exc:`~email.errors.HeaderParseError` is raised if the message object has no :mailheader:`Content-Type` header.

Note that using this method is subtly different from deleting the old :mailheader:`Content-Type` header and adding a new one with the new boundary via :meth:`add\_header`, because :meth:`set\_boundary` preserves the order of the :mailheader:`Content-Type` header in the list of headers.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 431)**

Unknown directive type "method".

```
.. method:: get_content_charset(failobj=None)
```

Return the ``charset`` parameter of the :mailheader:`Content-Type` header, coerced to lower case. If there is no :mailheader:`Content-Type` header, or if that header has no ``charset`` parameter, \*failobj\* is returned.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 438)**

Unknown directive type "method".

```
.. method:: get_charsets(failobj=None)
```

Return a list containing the character set names in the message. If the message is a :mimetype:`multipart`, then the list will contain one element for each subpart in the payload, otherwise, it will be a list of length 1.

Each item in the list will be a string which is the value of the ``charset`` parameter in the :mailheader:`Content-Type` header for the represented subpart. If the subpart has no :mailheader:`Content-Type` header, no ``charset`` parameter, or is not of the :mimetype:`text` main MIME type, then that item in the returned list will be \*failobj\*.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 451)**

Unknown directive type "method".

```
.. method:: is_attachment
```

Return ``True`` if there is a :mailheader:`Content-Disposition` header and its (case insensitive) value is ``attachment``, ``False`` otherwise.

```
.. versionchanged:: 3.4.2
   is_attachment is now a method instead of a property, for consistency
   with :meth:`~email.message.Message.is_multipart`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 461)

Unknown directive type "method".

```
.. method:: get_content_disposition()
```

Return the lowercased value (without parameters) of the message's :mailheader:`Content-Disposition` header if it has one, or ``None``. The possible values for this method are \*inline\*, \*attachment\* or ``None`` if the message follows :rfc:`2183`.

```
.. versionadded:: 3.5
```

The following methods relate to interrogating and manipulating the content (payload) of the message.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 475)**

Unknown directive type "method".

```
.. method:: walk()
```

The :meth:`walk` method is an all-purpose generator which can be used to iterate over all the parts and subparts of a message object tree, in depth-first traversal order. You will typically use :meth:`walk` as the iterator in a ``for`` loop; each iteration returns the next subpart.

Here's an example that prints the MIME type of every part of a multipart message structure:

```
.. testsetup::
```

```
from email import message_from_binary_file
with open('../Lib/test/test_email/data/msg_16.txt', 'rb') as f:
    msg = message_from_binary_file(f)
```

```
.. doctest::
```

```
>>> for part in msg.walk():
...     print(part.get_content_type())
multipart/report
text/plain
message/delivery-status
text/plain
text/plain
message/rfc822
text/plain
```

``walk`` iterates over the subparts of any part where :meth:`is\_multipart` returns ``True``, even though ``msg.get\_content\_maintype() == 'multipart'`` may return ``False``. We can see this in our example by making use of the ``\_structure`` debug helper function:

```
.. doctest::
```

```
>>> from email.iterators import _structure
>>> for part in msg.walk():
...     print(part.get_content_maintype() == 'multipart',
...           part.is_multipart())
True True
False False
False True
False False
False False
False True
False False
>>> _structure(msg)
multipart/report
  text/plain
  message/delivery-status
    text/plain
    text/plain
  message/rfc822
    text/plain
```

Here the ``message`` parts are not ``multiparts``, but they do contain subparts. ``is\_multipart()`` returns ``True`` and ``walk`` descends

into the subparts.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 536)**

Unknown directive type "method".

```
.. method:: get_body(preferencelist=('related', 'html', 'plain'))
```

Return the MIME part that is the best candidate to be the "body" of the message.

\*preferencelist\* must be a sequence of strings from the set ``related``, ``html``, and ``plain``, and indicates the order of preference for the content type of the part returned.

Start looking for candidate matches with the object on which the ``get\_body`` method is called.

If ``related`` is not included in \*preferencelist\*, consider the root part (or subpart of the root part) of any related encountered as a candidate if the (sub-)part matches a preference.

When encountering a ``multipart/related``, check the ``start`` parameter and if a part with a matching :mailheader: `Content-ID` is found, consider only it when looking for candidate matches. Otherwise consider only the first (default root) part of the ``multipart/related``.

If a part has a :mailheader: `Content-Disposition` header, only consider the part a candidate match if the value of the header is ``inline``.

If none of the candidates matches any of the preferences in \*preferencelist\*, return ``None``.

Notes: (1) For most applications the only \*preferencelist\* combinations that really make sense are ``('plain',)`` , ``('html', 'plain')`` , and the default ``('related', 'html', 'plain')`` . (2) Because matching starts with the object on which ``get\_body`` is called, calling ``get\_body`` on a ``multipart/related`` will return the object itself unless \*preferencelist\* has a non-default value. (3) Messages (or message parts) that do not specify a :mailheader: `Content-Type` or whose :mailheader: `Content-Type` header is invalid will be treated as if they are of type ``text/plain``, which may occasionally cause ``get\_body`` to return unexpected results.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 575)**

Unknown directive type "method".

```
.. method:: iter_attachments()
```

Return an iterator over all of the immediate sub-parts of the message that are not candidate "body" parts. That is, skip the first occurrence of each of ``text/plain``, ``text/html``, ``multipart/related``, or ``multipart/alternative`` (unless they are explicitly marked as attachments via :mailheader: `Content-Disposition: attachment`), and return all remaining parts. When applied directly to a ``multipart/related``, return an iterator over the all the related parts except the root part (ie: the part pointed to by the ``start`` parameter, or the first part if there is no ``start`` parameter or the ``start`` parameter doesn't match the :mailheader: `Content-ID` of any of the parts). When applied directly to a ``multipart/alternative`` or a non-``multipart``, return an empty iterator.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 591)**

Unknown directive type "method".

```
.. method:: iter_parts()
```

Return an iterator over all of the immediate sub-parts of the message, which will be empty for a non-`multipart`. (See also :meth:`~email.message.EmailMessage.walk`.)

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 598)**

Unknown directive type "method".

```
.. method:: get_content(*args, content_manager=None, **kw)
```

Call the :meth:`~email.contentmanager.ContentManager.get\_content` method of the `*content_manager*`, passing self as the message object, and passing along any other arguments or keywords as additional arguments. If `*content_manager*` is not specified, use the `content_manager` specified by the current :mod:`~email.policy`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 607)**

Unknown directive type "method".

```
.. method:: set_content(*args, content_manager=None, **kw)
```

Call the :meth:`~email.contentmanager.ContentManager.set\_content` method of the `*content_manager*`, passing self as the message object, and passing along any other arguments or keywords as additional arguments. If `*content_manager*` is not specified, use the `content_manager` specified by the current :mod:`~email.policy`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 616)**

Unknown directive type "method".

```
.. method:: make_related(boundary=None)
```

Convert a non-`multipart` message into a `multipart/related` message, moving any existing :mailheader: `Content-` headers and payload into a (new) first part of the `multipart`. If `*boundary*` is specified, use it as the boundary string in the multipart, otherwise leave the boundary to be automatically created when it is needed (for example, when the message is serialized).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 626)**

Unknown directive type "method".

```
.. method:: make_alternative(boundary=None)
```

Convert a non-`multipart` or a `multipart/related` into a `multipart/alternative`, moving any existing :mailheader: `Content-` headers and payload into a (new) first part of the `multipart`. If `*boundary*` is specified, use it as the boundary string in the multipart, otherwise leave the boundary to be automatically created when it is needed (for example, when the message is serialized).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 636)**

Unknown directive type "method".

```
.. method:: make_mixed(boundary=None)
```

Convert a non-`multipart`, a `multipart/related`, or a

`multipart-alternative` into a `multipart/mixed`, moving any existing :mailheader:`Content-` headers and payload into a (new) first part of the `multipart`. If \*boundary\* is specified, use it as the boundary string in the multipart, otherwise leave the boundary to be automatically created when it is needed (for example, when the message is serialized).

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 646)**

Unknown directive type "method".

```
.. method:: add_related(*args, content_manager=None, **kw)
```

If the message is a `multipart/related`, create a new message object, pass all of the arguments to its :meth:`set\_content` method, and :meth:`~email.message.Message.attach` it to the `multipart`. If the message is a non-`multipart`, call :meth:`make\_related` and then proceed as above. If the message is any other type of `multipart`, raise a :exc:`TypeError`. If \*content\_manager\* is not specified, use the `content\_manager` specified by the current :mod:`~email.policy`. If the added part has no :mailheader:`Content-Disposition` header, add one with the value `inline`.

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

Unknown directive type "method".

```
.. method:: add_alternative(*args, content_manager=None, **kw)
```

If the message is a `multipart/alternative`, create a new message object, pass all of the arguments to its :meth:`set\_content` method, and :meth:`~email.message.Message.attach` it to the `multipart`. If the message is a non-`multipart` or `multipart/related`, call :meth:`make\_alternative` and then proceed as above. If the message is any other type of `multipart`, raise a :exc:`TypeError`. If \*content\_manager\* is not specified, use the `content\_manager` specified by the current :mod:`~email.policy`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 671)**

Unknown directive type "method".

```
.. method:: add_attachment(*args, content_manager=None, **kw)
```

If the message is a `multipart/mixed`, create a new message object, pass all of the arguments to its :meth:`set\_content` method, and :meth:`~email.message.Message.attach` it to the `multipart`. If the message is a non-`multipart`, `multipart/related`, or `multipart-alternative`, call :meth:`make\_mixed` and then proceed as above. If \*content\_manager\* is not specified, use the `content\_manager` specified by the current :mod:`~email.policy`. If the added part has no :mailheader:`Content-Disposition` header, add one with the value `attachment`. This method can be used both for explicit attachments (:meth:`~email.message.Message.attach` with :mailheader:`Content-Disposition: attachment`) and `inline` attachments (:meth:`~email.message.Message.attach` with :mailheader:`Content-Disposition: inline`), by passing appropriate options to the `content\_manager`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ [cpython-main] [Doc] [library]email.message.rst, line 687)**

Unknown directive type "method".

```
.. method:: clear()
```

Remove the payload and all of the headers.

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

Unknown directive type "method".

```
.. method:: clear_content()
```

Remove the payload and all of the :exc:`Content-` headers, leaving all other headers intact and in their original order.

:class:`EmailMessage` objects have the following instance attributes:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 698); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 701)**

Unknown directive type "attribute".

```
.. attribute:: preamble
```

The format of a MIME document allows for some text between the blank line following the headers, and the first multipart boundary string. Normally, this text is never visible in a MIME-aware mail reader because it falls outside the standard MIME armor. However, when viewing the raw text of the message, or when viewing the message in a non-MIME aware reader, this text can become visible.

The *\*preamble\** attribute contains this leading extra-armor text for MIME documents. When the :class:`~email.parser.Parser` discovers some text after the headers but before the first boundary string, it assigns this text to the message's *\*preamble\** attribute. When the :class:`~email.generator.Generator` is writing out the plain text representation of a MIME message, and it finds the message has a *\*preamble\** attribute, it will write this text in the area between the headers and the first boundary. See :mod:`~email.parser` and :mod:`~email.generator` for details.

Note that if the message object has no preamble, the *\*preamble\** attribute will be `None`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 724)**

Unknown directive type "attribute".

```
.. attribute:: epilogue
```

The *\*epilogue\** attribute acts the same way as the *\*preamble\** attribute, except that it contains text that appears between the last boundary and the end of the message. As with the :attr:`~EmailMessage.preamble`, if there is no epilog text this attribute will be `None`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 732)**

Unknown directive type "attribute".

```
.. attribute:: defects
```

The *\*defects\** attribute contains a list of all the problems found when parsing this message. See :mod:`~email.errors` for a detailed description of the possible parsing defects.

This class represents a subpart of a MIME message. It is identical to :class:`EmailMessage`, except that no :mailheader:`MIME-

`Version`` headers are added when `.meth:~EmailMessage.set_content`` is called, since sub-parts do not need their own `mailheader:MIME-Version`` headers.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 741); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 741); [backlink](#)

Unknown interpreted text role "mailheader".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 741); [backlink](#)

Unknown interpreted text role "meth".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 741); [backlink](#)

Unknown interpreted text role "mailheader".

## Footnotes

- [1] Originally added in 3.4 as a `.term`provisional module <provisional package>``. Docs for legacy message class moved to `ref`compat32_message``.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 749); [backlink](#)

Unknown interpreted text role "term".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]email.message.rst, line 749); [backlink](#)

Unknown interpreted text role "ref".