NAME

Mail::Message::Dummy − a placeholder for a missing messages

INHERITANCE

```
Mail::Message::Dummy
  is a Mail::Message
  is a Mail::Reporter
```

SYNOPSIS

DESCRIPTION

Dummy messages are used by modules which maintain ordered lists of messages, usually based on message-id. A good example is Mail::Box::Thread::Manager, which detects related messages by scanning the known message headers for references to other messages. As long as the referenced messages are not found inside the mailbox, their place is occupied by a dummy.

Be careful when using modules which may create dummies. Before trying to access the header or body use **isDummy()** to check if the message is a dummy message.

Extends ‘‘DESCRIPTION’’ in Mail::Message.

METHODS

Extends ‘‘METHODS’’ in Mail::Message.

Constructors

Extends ‘‘Constructors’’ in Mail::Message.

`$obj`−>**clone**(%options)

Inherited, see ‘‘Constructors’’ in Mail::Message

Mail::Message::Dummy−>**new**($message_id, `%options`)

Create a new dummy message to occupy the space for a real message with the specified `$message_id`.

```
 -Option    --Defined in     --Default
  body        Mail::Message    <not used>
  body_type   Mail::Message    Mail::Message::Body::Lines
  deleted     Mail::Message    <false>
  field_type  Mail::Message    <not used>
  head        Mail::Message    <not used>
  head_type   Mail::Message    <not used>
  labels      Mail::Message    {}
  log         Mail::Reporter   'WARNINGS'
  messageId   Mail::Message    <required>
  modified    Mail::Message    <always false>
  trace       Mail::Reporter   'WARNINGS'
  trusted     Mail::Message    <always true>
```

body => OBJECT
body_type => CLASS
deleted => BOOLEAN
field_type => CLASS
head => OBJECT
head_type => CLASS
labels => ARRAY|HASH
log => LEVEL
messageId => STRING
modified => BOOLEAN
trace => LEVEL

trusted => BOOLEAN

example:

```
my $message = Mail::Message::Dummy->new($msgid);
if($message->isDummy) {...}
```

Constructing a message

Extends ‘‘Constructing a message’’ in Mail::Message.

`$obj`->**bounce**([<$rg_object|%options>])
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Bounce

Mail::Message::Dummy->**build**([$message|$part|$body], `$content`)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Build

Mail::Message::Dummy->**buildFromBody**($body, [$head], `$headers`)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Build

`$obj`->**forward**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardAttach**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardEncapsulate**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardInline**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardNo**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardPostlude**()
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardPrelude**()
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

`$obj`->**forwardSubject**(STRING)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Forward

Mail::Message::Dummy->**read**($fh|STRING|SCALAR|ARRAY, `%options`)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Read

`$obj`->**rebuild**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Rebuild

`$obj`->**reply**(%options)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Reply

`$obj`->**replyPrelude**([STRING|$field|$address|ARRAY-$of-$things])
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Reply

`$obj`->**replySubject**(STRING)
Mail::Message::Dummy->**replySubject**(STRING)
Inherited, see ‘‘Constructing a message’’ in Mail::Message::Construct::Reply

The message

Extends ‘‘The message’’ in Mail::Message.

`$obj`->**container**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**isDummy**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**isPart**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**messageId**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**partNumber**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**print**([$fh])
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**send**([$mailer], `%options`)
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**size**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**toplevel**()
Inherited, see ‘‘The message’’ in Mail::Message

`$obj->`**write**([$fh])
Inherited, see ‘‘The message’’ in Mail::Message

The header
Extends ‘‘The header’’ in Mail::Message.

`$obj->`**bcc**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**cc**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**date**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**destinations**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**from**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**get**($fieldname)
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**guessTimestamp**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**head**(...)
`$obj->`**nrLines**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**sender**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**study**($fieldname)
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**subject**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**timestamp**()
Inherited, see ‘‘The header’’ in Mail::Message

`$obj->`**to**()
Inherited, see ‘‘The header’’ in Mail::Message

The body
Extends ‘‘The body’’ in Mail::Message.

`$obj->`**body**([$body])
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**contentType**()
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**decoded**(%options)
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**encode**(%options)
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**isMultipart**()
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**isNested**()
Inherited, see ‘‘The body’’ in Mail::Message

`$obj->`**parts**([<’ALL’|’ACTIVE’|’DELETED’|’RECURSE’|$filter>])
Inherited, see ‘‘The body’’ in Mail::Message

Flags
Extends ‘‘Flags’’ in Mail::Message.

`$obj->`**delete**()
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**deleted**([BOOLEAN])
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**isDeleted**()
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**isModified**()
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**label**($label|PAIRS)
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**labels**()
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**labelsToStatus**()
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**modified**([BOOLEAN])
Inherited, see ‘‘Flags’’ in Mail::Message

`$obj->`**statusToLabels**()
Inherited, see ‘‘Flags’’ in Mail::Message

The whole message as text
Extends ‘‘The whole message as text’’ in Mail::Message.

`$obj->`**file**()
Inherited, see ‘‘The whole message as text’’ in Mail::Message::Construct::Text

`$obj->`**lines**()
Inherited, see ‘‘The whole message as text’’ in Mail::Message::Construct::Text

`$obj->`**printStructure**([$fh|undef],[$indent])
Inherited, see ‘‘The whole message as text’’ in Mail::Message::Construct::Text

`$obj->`**string**()
Inherited, see ‘‘The whole message as text’’ in Mail::Message::Construct::Text

Internals

Extends ‘‘Internals’’ in Mail::Message.

`$obj->`**clonedFrom**()
Inherited, see ‘‘Internals’’ in Mail::Message

Mail::Message::Dummy->**coerce**($message, `%options`)
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**isDelayed**()
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**readBody**(`$parser`, `$head`, [$bodytype])
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**readFromParser**(`$parser`, [$bodytype])
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**readHead**(`$parser`, [$class])
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**recursiveRebuildPart**($part, `%options`)
Inherited, see ‘‘Internals’’ in Mail::Message::Construct::Rebuild

`$obj->`**storeBody**($body)
Inherited, see ‘‘Internals’’ in Mail::Message

`$obj->`**takeMessageId**([STRING])
Inherited, see ‘‘Internals’’ in Mail::Message

Error handling

Extends ‘‘Error handling’’ in Mail::Message.

`$obj->`**AUTOLOAD**()
Inherited, see ‘‘METHODS’’ in Mail::Message::Construct

`$obj->`**addReport**($object)
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Mail::Message::Dummy->**defaultTrace**([$level]|[$loglevel, `$tracelevel`]|[$level, `$callback`])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**errors**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**log**([$level, [$strings]])
Mail::Message::Dummy->**log**([$level, [$strings]])
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**logPriority**($level)
Mail::Message::Dummy->**logPriority**($level)
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**logSettings**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj->`**notImplemented**()
Inherited, see ‘‘Error handling’’ in Mail::Reporter

`$obj`->**report**([$level])
Inherited, see ''Error handling'' in Mail::Reporter

`$obj`->**reportAll**([$level])
Inherited, see ''Error handling'' in Mail::Reporter

`$obj`->**shortSize**([$value])
Mail::Message::Dummy->**shortSize**([$value])
Inherited, see ''Error handling'' in Mail::Message

`$obj`->**shortString**()
Inherited, see ''Error handling'' in Mail::Message

`$obj`->**trace**([$level])
Inherited, see ''Error handling'' in Mail::Reporter

`$obj`->**warnings**()
Inherited, see ''Error handling'' in Mail::Reporter

Cleanup

Extends ''Cleanup'' in Mail::Message.

`$obj`->**DESTROY**()
Inherited, see ''Cleanup'' in Mail::Reporter

`$obj`->**destruct**()
Inherited, see ''Cleanup'' in Mail::Message

DETAILS

Extends ''DETAILS'' in Mail::Message.

DIAGNOSTICS

Error: Cannot coerce a `$class` object into a `$class` object

Error: Cannot include forward source as `$include`.
Unknown alternative for the forward(include). Valid choices are `NO`, `INLINE`, `ATTACH`, and `ENCAPSULATE`.

Error: Cannot include reply source as `$include`.
Unknown alternative for the `include` option of **reply()**. Valid choices are `NO`, `INLINE`, and `ATTACH`.

Error: Message-Id is required for a dummy.
A dummy message occupies the place for a real message. When a dummy is created, the id of the message which place it is holding must be known.

Error: Method bounce requires To, Cc, or Bcc
The message **bounce()** method forwards a received message off to someone else without modification; you must specified it's new destination. If you have the urge not to specify any destination, you probably are looking for **reply()**. When you wish to modify the content, use **forward()**.

Error: Method forwardAttach requires a preamble

Error: Method forwardEncapsulate requires a preamble

Error: No address to create forwarded to.
If a forward message is created, a destination address must be specified.

Error: No default mailer found to send message.
The message **send()** mechanism had not enough information to automatically find a mail transfer agent to sent this message. Specify a mailer explicitly using the `via` options.

Error: No rebuild rule `$name` defined.

Error: Only **build()** Mail::Message's; they are not in a folder yet
You may wish to construct a message to be stored in a some kind of folder, but you need to do that in two steps. First, create a normal Mail::Message, and then add it to the folder. During this **Mail::Box::addMessage()** process, the message will get **coerce()**-d into the right message type,

adding storage information and the like.

Error: Package `$package` does not implement `$method`.
: Fatal error: the specific package (or one of its superclasses) does not implement this method where it should. This message means that some other related classes do implement this method however the class at hand does not. Probably you should investigate this and probably inform the author of the package.

Error: You cannot take the head/body of a dummy message
: Dummy messages are place-holders in message threads: the thread detected the existence of the message, because it found the message-id in a Reply-To or References field, however it did not find the header and body of the message yet. Use **isDummy()** to check whether the thread node returned a dummy or not.

Error: coercion starts with some object

SEE ALSO

This module is part of Mail-Box distribution version 3.009, built on August 18, 2020. Website: *http://perl.overmeer.net/CPAN/*

LICENSE