

NAME

Mail::Message::Field::Addresses – Fields with e-mail addresses

INHERITANCE

```
Mail::Message::Field::Addresses
  is a Mail::Message::Field::Structured
  is a Mail::Message::Field::Full
  is a Mail::Message::Field
  is a Mail::Reporter
```

SYNOPSIS

```
my $cc = Mail::Message::Field::Full->new('Cc');
my $me = Mail::Message::Field::Address->parse('"Test" <test@mail.box>')
    or die;

my $other = Mail::Message::Field::Address->new(phrase => 'Other'
    , address => 'other@example.com')
    or die;

$cc->addAddress($me);
$cc->addAddress($other, group => 'them');
$cc->addAddress(phrase => 'third', address => 'more@any.museum'
    , group => 'them');

my $group = $cc->addGroup(name => 'colleagues');
$group->addAddress($me);
$group->addAddress(phrase => "You", address => 'you@example.com');

my $msg = Mail::Message->build(Cc => $cc);
print $msg->string;

my $g = Mail::Message::Field::AddrGroup->new(...);
$cc->addGroup($g);
```

DESCRIPTION

All header fields which contain e-mail addresses only. Not all address fields have the same possibilities, but they are all parsed the same: you never know how broken the applications are which produce those messages.

When you try to create constructs which are not allowed for a certain kind of field, you will be warned.

Extends “DESCRIPTION” in Mail::Message::Field::Structured.

OVERLOADED

Extends “OVERLOADED” in Mail::Message::Field::Structured.

overload: “”

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: 0+

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: <=>

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: bool

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: cmp

Inherited, see “OVERLOADED” in Mail::Message::Field

overload: **stringification**

Inherited, see “OVERLOADED” in Mail::Message::Field::Full

METHODS

Extends “METHODS” in Mail::Message::Field::Structured.

Constructors

Extends “Constructors” in Mail::Message::Field::Structured.

`$obj->clone()`

Inherited, see “Constructors” in Mail::Message::Field

`Mail::Message::Field::Addresses->from($field, %options)`

Inherited, see “Constructors” in Mail::Message::Field::Full

`Mail::Message::Field::Addresses->new()`

--Option	--Defined in	--Default
<code>attributes</code>	<code>Mail::Message::Field::Structured</code>	<code><ignored></code>
<code>charset</code>	<code>Mail::Message::Field::Full</code>	<code>undef</code>
<code>datum</code>	<code>Mail::Message::Field::Structured</code>	<code>undef</code>
<code>encoding</code>	<code>Mail::Message::Field::Full</code>	<code>'q'</code>
<code>force</code>	<code>Mail::Message::Field::Full</code>	<code>false</code>
<code>language</code>	<code>Mail::Message::Field::Full</code>	<code>undef</code>
<code>log</code>	<code>Mail::Reporter</code>	<code>'WARNINGS'</code>
<code>trace</code>	<code>Mail::Reporter</code>	<code>'WARNINGS'</code>

`attributes => ATTRS`

`charset => STRING`

`datum => STRING`

`encoding => 'q'|'Q'|'b'|'B'`

`force => BOOLEAN`

`language => STRING`

`log => LEVEL`

`trace => LEVEL`

The field

Extends “The field” in Mail::Message::Field::Structured.

`$obj->isStructured()`

`Mail::Message::Field::Addresses->isStructured()`

Inherited, see “The field” in Mail::Message::Field

`$obj->length()`

Inherited, see “The field” in Mail::Message::Field

`$obj->nrLines()`

Inherited, see “The field” in Mail::Message::Field

`$obj->print([$fh])`

Inherited, see “The field” in Mail::Message::Field

`$obj->size()`

Inherited, see “The field” in Mail::Message::Field

`$obj->string([$wrap])`

Inherited, see “The field” in Mail::Message::Field

`$obj->toDisclose()`

Inherited, see “The field” in Mail::Message::Field

Access to the name

Extends “Access to the name” in Mail::Message::Field::Structured.

`$obj->Name()`

Inherited, see “Access to the name” in Mail::Message::Field

`$obj->name()`

Inherited, see “Access to the name” in Mail::Message::Field

`$obj->wellformedName([STRING])`

Inherited, see “Access to the name” in Mail::Message::Field

Access to the body

Extends “Access to the body” in Mail::Message::Field::Structured.

`$obj->body()`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->decodedBody(%options)`

Inherited, see “Access to the body” in Mail::Message::Field::Full

`$obj->folded()`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->foldedBody([$body])`

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->stripCFWS([STRING])`

Mail::Message::Field::Addresses->stripCFWS([STRING])

Inherited, see “Access to the body” in Mail::Message::Field

`$obj->unfoldedBody([$body, [$wrap]])`

Inherited, see “Access to the body” in Mail::Message::Field

Access to the content

Extends “Access to the content” in Mail::Message::Field::Structured.

`$obj->addAddress([$address], %options)`

Add an `$address` to the field. The addresses are organized in groups. If no group is specified, the default group is taken to store the address in. If no `$address` is specified, the option must be sufficient to create a Mail::Message::Field::Address from. See the `%options` of Mail::Message::Field::Address::new().

```
-Option--Default
group    ''
```

```
group => STRING
```

`$obj->addAttribute(...)`

Attributes are not supported for address fields.

`$obj->addGroup($group|%options)`

Add a group of addresses to this field. A `$group` can be specified, which is a Mail::Message::Field::AddrGroup object, or one is created for you using the `%options`. The group is returned.

```
-Option--Default
name    ''
```

```
name => STRING
```

`$obj->addresses()`

Returns a list with all addresses defined in any group of addresses: all addresses which are specified on the line. The addresses are Mail::Message::Field::Address objects.

example:

```
my @addr = $field->addresses;
```

`$obj->attrPairs()`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->attribute($object|<STRING, %options>|<$name,$value,%options>)`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->attributes()`

Inherited, see “Access to the content” in Mail::Message::Field::Structured

`$obj->beautify()`

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->comment([STRING])`

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->createComment(STRING, %options)`

Mail::Message::Field::Addresses->**createComment**(STRING, %options)

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->createPhrase(STRING, %options)`

Mail::Message::Field::Addresses->**createPhrase**(STRING, %options)

Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->group($name)`

Returns the group of addresses with the specified \$name, or undef if it does not exist. If \$name is undef, then the default groep is returned.

`$obj->groupNames()`

Returns a list with all group names which are defined.

`$obj->groups()`

Returns all address groups which are defined in this field. Each element is a Mail::Message::Field::AddrGroup object.

`$obj->study()`

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->toDate([$time])`

Mail::Message::Field::Addresses->**toDate**([\$time])

Inherited, see “Access to the content” in Mail::Message::Field

`$obj->toInt()`

Inherited, see “Access to the content” in Mail::Message::Field

Other methods

Extends “Other methods” in Mail::Message::Field::Structured.

`$obj->dateToTimestamp(STRING)`

Mail::Message::Field::Addresses->**dateToTimestamp**(STRING)

Inherited, see “Other methods” in Mail::Message::Field

Internals

Extends “Internals” in Mail::Message::Field::Structured.

`$obj->consume($line | <$name,<$body|$objects>>)`

Inherited, see “Internals” in Mail::Message::Field

`$obj->decode(STRING, %options)`

Mail::Message::Field::Addresses->**decode**(STRING, %options)

Inherited, see “Internals” in Mail::Message::Field::Full

`$obj->defaultWrapLength([$length])`

Inherited, see “Internals” in Mail::Message::Field

`$obj->encode(STRING, %options)`
 Inherited, see “Internals” in Mail::Message::Field::Full

`$obj->fold($name, $body, [$maxchars])`
 Mail::Message::Field::Addresses->**fold**(\$name, \$body, [\$maxchars])
 Inherited, see “Internals” in Mail::Message::Field

`$obj->setWrapLength([$length])`
 Inherited, see “Internals” in Mail::Message::Field

`$obj->stringifyData(STRING|ARRAY|Objects)`
 Inherited, see “Internals” in Mail::Message::Field

`$obj->unfold(STRING)`
 Inherited, see “Internals” in Mail::Message::Field

Parsing

Extends “Parsing” in Mail::Message::Field::Structured.

`$obj->consumeAddress(STRING, %options)`
 Try to destilate address information from the STRING. Returned are an address**object** and the left-over string. If no address was found, the first returned value is undef.

`$obj->consumeComment(STRING)`
 Mail::Message::Field::Addresses->**consumeComment**(STRING)
 Inherited, see “Parsing” in Mail::Message::Field::Full

`$obj->consumeDomain(STRING)`
 Try to get a valid domain representation from STRING. Returned are the domain string as found (or undef) and the rest of the string.

`$obj->consumeDotAtom(STRING)`
 Inherited, see “Parsing” in Mail::Message::Field::Full

`$obj->consumePhrase(STRING)`
 Mail::Message::Field::Addresses->**consumePhrase**(STRING)
 Inherited, see “Parsing” in Mail::Message::Field::Full

`$obj->datum([$value])`
 Inherited, see “Parsing” in Mail::Message::Field::Structured

`$obj->parse(STRING)`
 Inherited, see “Parsing” in Mail::Message::Field::Full

`$obj->produceBody()`
 Inherited, see “Parsing” in Mail::Message::Field::Full

Error handling

Extends “Error handling” in Mail::Message::Field::Structured.

`$obj->AUTOLOAD()`
 Inherited, see “Error handling” in Mail::Reporter

`$obj->addReport($object)`
 Inherited, see “Error handling” in Mail::Reporter

`$obj->defaultTrace([$level][[$loglevel, $tracelevel]][[$level, $callback]])`
 Mail::Message::Field::Addresses->**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::Field::Addresses->log([\$level, [\$strings]])
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`
 Mail::Message::Field::Addresses->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

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

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

Cleanup

Extends “Cleanup” in Mail::Message::Field::Structured.

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

DETAILS

Extends “DETAILS” in Mail::Message::Field::Structured.

DIAGNOSTICS

Warning: Field content is not numerical: `$content`
 The numeric value of a field is requested (for instance the `Lines` or `Content-Length` fields should be numerical), however the data contains weird characters.

Warning: Illegal character in charset '`$charset`'
 The field is created with an utf8 string which only contains data from the specified character set. However, that character set can never be a valid name because it contains characters which are not permitted.

Warning: Illegal character in field name `$name`
 A new field is being created which does contain characters not permitted by the RFCs. Using this field in messages may break other e-mail clients or transfer agents, and therefore mutilate or extinguish your message.

Warning: Illegal character in language '`$lang`'
 The field is created with data which is specified to be in a certain language, however, the name of the language cannot be valid: it contains characters which are not permitted by the RFCs.

Warning: Illegal encoding '`$encoding`', used '`q`'
 The RFCs only permit base64 (`b` or `B`) or quoted-printable (`q` or `Q`) encoding. Other than these four options are illegal.

Error: No attributes for address fields.
 Is is not possible to add attributes to address fields: it is not permitted by the RFCs.

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.

SEE ALSO

This module is part of Mail-Message distribution version 3.012, built on February 11, 2022. Website:
<http://perl.overmeer.net/CPAN/>

LICENSE

Copyrights 2001–2022 by [Mark Overmeer <markov@cpan.org>]. For other contributors see ChangeLog.

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.
See *<http://dev.perl.org/licenses/>*