

**NAME**

Mail::Message::Field::Fast – one line of a message header

**INHERITANCE**

```
Mail::Message::Field::Fast
    is a Mail::Message::Field
    is a Mail::Reporter
```

**SYNOPSIS**

See Mail::Message::Field

**DESCRIPTION**

This is the faster, but less flexible implementation of a header field. The data is stored in an array, and some hacks are made to speeds things up. Be gentle with me, and consider that each message contains many of these lines, so speed is very important here.

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

**OVERLOADED**

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

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

**METHODS**

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

**Constructors**

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

\$obj->clone()

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

Mail::Message::Field::Fast->new(\$data)

The constructor of this object does not follow the usual practise within the Mail::Box suite: it does not use the constructor **Mail::Reporter::new()**. Therefor it has no logging or tracing facilities.

The method can be used in one of the following ways:

- **new** LINE

Pass a LINE as it could be found in a file: a (possibly folded) line which is terminated by a new-line.

- **new** NAME, (BODY|OBJECTS), [ATTRIBUTES]

A set of values which shape the line.

Create a new header field object. Specify the whole LINE at once, and it will be split-up for you. I case you already have the parts of the header line, you may specify them separately as NAME and BODY.

In case you specify a single OBJECT, or a reference to an array of OBJECTS, these objects are processed to become suitable to fill a field, usually by simple strification. When you specify one or more Mail::Address objects, these are transformed into a string using their `format` method. You may

also add one Mail::Message::Field, whose body is taken. In case of an array, the elements are joined into one string with a comma.

ATTRIBUTES can be exactly one string which may contain multiple attributes at once, quoted and formatted as required in RFC2822. As alternative, list of key-value pairs can be used. In this case, the values will get quoted if needed and everything formatted as the protocol demands.

```
--Option--Defined in      --Default
log      Mail::Reporter    <disabled>
trace    Mail::Reporter    <disabled>
```

log => LEVEL

trace => LEVEL

example:

```
my $mime = Mail::Message::Field->new(
    'Content-Type: text/plain; charset=US-ASCII');

my $mime = Mail::Message::Field->new(
    'Content-Type' => 'text/plain; charset=US-ASCII');

my $mime = Mail::Message::Field->new(
    'Content-Type' => 'text/plain', 'charset=US-ASCII');

my $mime = Mail::Message::Field->new(
    'Content-Type' => 'text/plain', charset => 'Latin1');

my $mime = Mail::Message::Field->new(
    To => Mail::Address->new('My', 'me@example.com'));

my $mime = Mail::Message::Field->new(
    Cc => [ Mail::Address->new('You', 'you@example.com')
           , Mail::Address->new('His', 'he@example.com')
         ] );
```

But in practice, you can simply call

```
my $head = Mail::Message::Head->new;
$head->add( 'Content-Type' => 'text/plain'
           , charset => 'utf8');
```

which implicitly calls this constructor (when needed). You can specify the same things for **Mail::Message::Head::Complete::add()** as this new accepts.

### The field

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

`$obj->isStructured()`

Mail::Message::Field::Fast->**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.

`$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.

`$obj->body()`  
Inherited, see “Access to the body” in Mail::Message::Field

`$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::Fast->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.

`$obj->addresses()`  
Inherited, see “Access to the content” in Mail::Message::Field

`$obj->attribute( $name, [$value] )`  
Inherited, see “Access to the content” in Mail::Message::Field

`$obj->attributes()`  
Inherited, see “Access to the content” in Mail::Message::Field

`$obj->comment( [STRING] )`  
Inherited, see “Access to the content” in Mail::Message::Field

`$obj->study()`  
Inherited, see “Access to the content” in Mail::Message::Field

`$obj->toDate( [$time] )`  
`Mail::Message::Field::Fast->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.

`$obj->dateToTimestamp(String)`

Mail::Message::Field::Fast->**dateToTimestamp**(String)

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

**Internals**

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

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

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

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

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

`$obj->fold( $name, $body, [$maxchars] )`

Mail::Message::Field::Fast->**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

**Error handling**

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

`$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::Fast->**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::Fast->**log**( [\$level, [\$strings]] )

Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`

Mail::Message::Field::Fast->**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.

`$obj->DESTROY()`

Inherited, see “Cleanup” in Mail::Reporter

### **DETAILS**

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

### **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 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.

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/>