

NAME

Mail::Message::Field::Structured – one line of a structured message header

INHERITANCE

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

```
Mail::Message::Field::Structured is extended by
Mail::Message::Field::Addresses
Mail::Message::Field::AuthResults
Mail::Message::Field::DKIM
Mail::Message::Field::Date
Mail::Message::Field::URIs
```

SYNOPSIS

```
my $f = Mail::Message::Field::Full
    ->new('Content-Type' => 'text/html');

my @encode = (charset => 'jp', use_continuations => 1);
$f->attribute('filename=passwd');
$f->attribute(filename => 'passwd', @encode);

my $attr = Mail::Message::Field::Attribute->new(...);
$f->attribute($attr);
```

DESCRIPTION

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

OVERLOADED

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

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::Full.

Constructors

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

`$obj->clone()`

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

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

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

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

- **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, OPTIONS**

A set of values which shape the line.

The NAME is a wellformed header name (you may use **wellformedName()**) to be sure about the casing. The BODY is a string, one object, or an ref-array of objects. In case of objects, they must fit to the constructor of the field: the types which are accepted may differ. The optional ATTRIBUTE list contains Mail::Message::Field::Attribute objects. Finally, there are some OPTIONS.

-Option	--Defined in	--Default
attributes		[]
charset	Mail::Message::Field::Full	undef
datum		undef
encoding	Mail::Message::Field::Full	'q'
force	Mail::Message::Field::Full	false
language	Mail::Message::Field::Full	undef
log	Mail::Reporter	'WARNINGS'
trace	Mail::Reporter	'WARNINGS'

attributes => ATTRS

There are various ways to specify these attributes: pass a reference to an array which list of key-value pairs representing attributes, or reference to a hash containing these pairs, or an array with Mail::Message::Field::Attribute objects.

charset => STRING

datum => STRING

The method name *body* is very confusing, even in the RFC. In MailBox, for historical reasons, **body()** returns the part of the field contents before the first semi-colon. **foldedBody()** and **unfoldedBody()** address the whole field.

There is no common name for the piece of data before the parameters (attributes) in the field-content mentioned in the RFCs, so let's call it **datum**.

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

force => BOOLEAN

language => STRING

log => LEVEL

trace => LEVEL

example: of a structured field

```
my @attrs    = (Mail::Message::Field::Attribute->new(...), ...);
my @options = (extra => 'the color blue');
my $t = Mail::Message::Field::Full->new(To => \@addrs, @attrs, @options);
```

The field

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

\$obj->**isStructured()**

Mail::Message::Field::Structured->**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::Full.

`$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::Full.

`$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::Structured->`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::Full.

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

`$obj->attrPairs()`
 Returns a list with attribute name and value pairs.

`$obj->attribute($object|<STRING, %options>|<$name,$value,%options>)`
 Add an attribute to the field. The attributes are added left-to-right into the string representation of the field, although the order of the attributes is un-important, according to the RFCs.

You may pass a fully prepared Mail::Message::Field::Attribute `$object`, if you like to do all preparations for correct representation of the data yourself. You may also pass one STRING, which is a fully prepared attribute. This STRING will not be changed, so be careful about quoting and encodings.

As third possibility, you can specify an attribute `$name` and its `$value`. An attribute object will be

created for you implicitly in both cases where such object is not supplied, passing the `%options`. See **Mail::Message::Field::Attribute::new()** about the available `%options`.

The attribute object is returned, however, when continuations are used this may be an object you already know about. `undef` is returned when construction fails (when the attribute is incorrect).

example:

```
$f->attribute(filename => 'passwd');
$f->attribute(filename => 'passwd', use_continuations => 0);

my $attr = Mail::Message::Field::Attribute->new(...);
$f->attribute($attr);
```

\$obj->attributes()

Returns a list with all attributes, which are all `Mail::Message::Field::Attribute` objects. The attributes are not ordered in any way. The list may be empty. Double attributes or continuations are folded into one.

\$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::Structured->createComment(STRING, %options)

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

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

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

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

\$obj->study()

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

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

Mail::Message::Field::Structured->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::Full`.

\$obj->dateToTimestamp(STRING)

Mail::Message::Field::Structured->dateToTimestamp(STRING)

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

Internals

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

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

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

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

Mail::Message::Field::Structured->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::Structured->`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::Full.

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

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

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

`$obj->datum([$value])`
 The part of the field before the semi-colon (;).

`$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::Full.

`$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, $callback])`
 Mail::Message::Field::Structured->`defaultTrace([$level][[$loglevel, $tracelevel]][$level, $callback])`
 Inherited, see “Error handling” in Mail::Reporter

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

`$obj->log([$level, [$strings]])`
 Mail::Message::Field::Structured->`log([$level, [$strings]])`
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`
 Mail::Message::Field::Structured->`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::Full.

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

DETAILS

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

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: 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/>