

NAME

Mail::Message::Field::Attribute – one attribute of a full field

INHERITANCE

Mail::Message::Field::Attribute
is a Mail::Reporter

SYNOPSIS

```

my $field      = $msg->head->get('Content-Disposition') or return;
my $full       = $field->study;      # full understanding in unicode
my $filename   = $full->attribute('filename')           or return;

print ref $filename;      # this class name
print $filename;          # the attributes content in utf-8
print $filename->value;    # same
print $filename->string;   # print string as was found in the file
$filename->print(\*OUT);   # print as was found in the file

```

DESCRIPTION

Attributes within MIME fields can be quite complex, and therefore be slow and consumes a lot of memory. The Mail::Message::Field::Fast and Mail::Message::Field::Flex simplify them the attributes a lot, which may result in erroneous behavior in rare cases. With the increase of non-western languages on Internet, the need for the complex headers becomes more and more in demand.

A Mail::Message::Field::Attribute can be found in any structured Mail::Message::Field::Full header field.

Extends “DESCRIPTION” in Mail::Reporter.

OVERLOADED

overload: **comparison**

When the second argument is a field, then both attribute name (case-sensitive) and the decoded value must be the same. Otherwise, the value is compared.

overload: **stringification**

Returns the decoded content of the attribute.

METHODS

Extends “METHODS” in Mail::Reporter.

Constructors

Extends “Constructors” in Mail::Reporter.

Mail::Message::Field::Attribute->**new**(<\$name, [\$value] | STRING>, %options)

Create a new attribute \$name with the optional \$value. If no \$value is specified, the first argument of this method is inspected for an equals sign '='. If that character is present, the argument is taken as STRING, containing a preformatted attribute which is processed. Otherwise, the argument is taken as name without \$value: set the value later with **value()**.

Whether encoding takes place depends on the %options and the existence of non-ascii characters in the \$value. The \$name can only contain ascii characters, hence is never encoded.

To speed things up, attributes are not derived from the Mail::Reporter base-class.

| -Option | --Defined in | --Default |
|-------------------|----------------|------------|
| charset | | 'us-ascii' |
| language | | undef |
| log | Mail::Reporter | 'WARNINGS' |
| trace | Mail::Reporter | 'WARNINGS' |
| use_continuations | | <true> |

charset => STRING

The \$value is translated from utf-8 (Perl internal) to this character set, and the resulting string is encoded if required. us-ascii is the normal encoding for e-mail. Valid character sets can be found with Encode::encodings('all').

language => STRING

RFC2231 adds the possibility to specify a language with the field. When no language is specified, none is included in the encoding. Valid language names are defined by RFC2130. This module has only limited support for this feature.

log => LEVEL

trace => LEVEL

use_continuations => BOOLEAN

Continuations are used to break-up long parameters into pieces which are no longer than 76 characters. Encodings are specified in RFC2231, but not supported by some Mail User Agents.

example:

```
my $fn      = Mail::Message::Field::Attribute
              ->new(filename => 'xyz');

my $fattr = 'Mail::Message::Field::Attribute'; # abbrev
my $fn     = $fattr->new
( filename => "Re\xC7u"
, charset  => 'iso-8859-15'
, language => 'nl-BE'
);
print $fn;
# --> filename*=iso-8859-15'nl-BE'Re%C7u
```

Error handling

Extends “Error handling” in Mail::Reporter.

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

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

\$obj->**logPriority**(\$level)

Mail::Message::Field::Attribute->**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::Reporter.

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

The attribute

`$obj->addComponent(STRING)`
 A component is a parameter as defined by RFC2045, optionally using encoding or continuations as defined by RFC2231. Components of an attribute are found when a field is being parsed. The RFCs are very strict on valid characters, but we cannot be: you have to accept what is coming in if you can.

example:

```
my $param = Mail::Message::Field::Attribute->new;
$params->addComponent( "filename*=iso10646'nl-BE'%Re\47u" );
```

`$obj->charset()`
 Returns the character set which is used for this parameter. If any component is added which contains character set information, this is directly available. Be warned that a character-set is case insensitive.

`$obj->language()`
 Returns the language which is defined in the argument. If no language is defined `undef` is returned, which should be interpreted as “ANY”

`$obj->name()`
 Returns the name of this attribute.

`$obj->string()`
 Returns the parameter as reference to an array of lines. When only one line is returned, it may be short enough to fit on the same line with other components of the header field.

`$obj->value([STRING])`
 Returns the value of this parameter, optionally after setting it first.

Attribute encoding

`$obj->decode()`
 Translate all known continuations into a value. The produced value is returned and may be utf-8 encoded or a plain string.

`$obj->encode()`

Internals

`$obj->mergeComponent($attribute)`
 Merge the components from the specified attribute into this attribute. This is needed when components of the same attribute are created separately. Merging is required by the field parsing.

DIAGNOSTICS

Warning: Illegal character in parameter name '\$name'

The specified parameter name contains characters which are not permitted by the RFCs. You can better change the name into something which is accepted, or risk applications to corrupt or ignore the 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.

Error: Too late to merge: value already changed.

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/>