

**NAME**

Mail::Message::Field::DKIM – message header field for dkim signatures

**INHERITANCE**

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

**SYNOPSIS**

```
my $f = Mail::Message::Field->new('DKIM-Signature' => '...');

my $g = Mail::Message::Field->new('DKIM-Signature');
$g->add...
```

**DESCRIPTION**

Decode the information contained in a DKIM header. You can also construct DKIM-Signature headers this way. However, verification and signing is not yet implemented.

This implementation is based on RFC6376.

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::DKIM->from($field, %options)`

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

`Mail::Message::Field::DKIM->new($data)`

--Option	--Defined in	--Default
attributes	Mail::Message::Field::Structured	<ignored>
charset	Mail::Message::Field::Full	undef
datum	Mail::Message::Field::Structured	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

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::DKIM->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::DKIM->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->addAttribute(...)`  
 Attributes are not supported here.

`$obj->addTag($name, $value|@values)`  
 Add a tag to the set. When the tag already exists, it is replaced. Names are (converted to) lower-case. When multiple values are given, they will be concatenated with a blank (and may get folded there later)

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

`$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::DKIM->createComment(STRING, %options)  
 Inherited, see “Access to the content” in Mail::Message::Field::Full

`$obj->createPhrase(STRING, %options)`  
 Mail::Message::Field::DKIM->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

`$obj->tag($name)`  
 Returns the value for the named tag.

`$obj->toDate( [$time] )`  
 Mail::Message::Field::DKIM->toDate( [\$time] )  
 Inherited, see “Access to the content” in Mail::Message::Field

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

*DKIM-Signature tags*

The tag methods return the tag-value content without any validation or modification. For many situations, the actual content does not need (expensive) validation and interpretation.

`$obj->tagAgentID()`

The Agent or User Identifier (AUID). Defaults to `@$domain`

`$obj->tagAlgorithm()`

Signature algorithm. Should be `rsa-sha(1|256)`; check before use. Required.

`$obj->tagBodyLength()`

The number of octets which were used to calculate the hash. By default, the whole body was used.

`$obj->tagC14N()`

The canonicalization method used. Defaults to `'simple/simple'`.

`$obj->tagDomain()`

The sub-domain (SDID) which claims responsibility for this signature. Required.

`$obj->tagExpires()`

The timestamp when the signature will expire. Recommended.

`$obj->tagExtract()`

Some headers from the original message packed together.

`$obj->tagQueryMethods()`

A colon-separated list of method which can be used to retrieve the public key. The default is `"dns/txt"` (currently the only valid option)

`$obj->tagSelector()`

The selector subdividing the `$domain` tag. Required.

`$obj->tagSignData()`

`$obj->tagSignature()`

Message signature in base64, with whitespaces removed. Required.

`$obj->tagSignedHeaders()`

The colon separated list of headers which need to be included in the signature. Required.

`$obj->tagTimestamp()`

When the signature was created in UNIX-like seconds (since 1970). Recommended.

`$obj->tagVersion()`

Signature header syntax version (usually 1)

**Other methods**

Extends "Other methods" in `Mail::Message::Field::Structured`.

`$obj->dateToTimestamp(String)`

`Mail::Message::Field::DKIM->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::DKIM->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::DKIM->**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->consumeComment(STRING)`  
 Mail::Message::Field::DKIM->**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::DKIM->**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::DKIM->**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::DKIM->**log**( [\$level, [\$strings]] )  
 Inherited, see “Error handling” in Mail::Reporter

`$obj->logPriority($level)`  
 Mail::Message::Field::DKIM->**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 DKIM headers

Is not possible to add attributes to this field.

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/>