NAME

Mail::Message::Field::Address − One e−mail address

INHERITANCE

```
Mail::Message::Field::Address
  is a Mail::Identity
  is a User::Identity::Item
```

SYNOPSIS

```
my $addr = Mail::Message::Field::Address->new(...);

my $ui   = User::Identity->new(...);
my $addr = Mail::Message::Field::Address->coerce($ui);

my $mi   = Mail::Identity->new(...);
my $addr = Mail::Message::Field::Address->coerce($mi);

print $addr->address;
print $addr->fullName;   # possibly unicode!
print $addr->domain;
```

DESCRIPTION

Many header fields can contain e−mail addresses. Each e−mail address can be represented by an object of this class. These objects will handle interpretation and character set encoding and decoding for you.

Extends ‘‘DESCRIPTION’’ in Mail::Identity.

OVERLOADED

overload: **boolean**

The object used as boolean will always return `true`

overload: **string** `$comparison`

Two address objects are the same when their email addresses are the same.

overload: **stringification**

When the object is used in string context, it will return the encoded representation of the e−mail address, just like **string()** does.

METHODS

Extends ‘‘METHODS’’ in Mail::Identity.

Constructors

Extends ‘‘Constructors’’ in Mail::Identity.

`$obj`−>**coerce**(<STRING|$object>, `%options`)

Try to coerce the `$object` into a `Mail::Message::Field::Address`. In case of a STRING, it is interpreted as an email address.

The `%options` are passed to the object creation, and overrule the values found in the `$object`. The result may be `undef` or a newly created object. If the `$object` is already of the correct type, it is returned unmodified.

The `$object` may currently be a Mail::Address, a Mail::Identity, or a User::Identity. In case of the latter, one of the user’s addresses is chosen at random.

Mail::Message::Field::Address−>**new**([$name], `%options`)

Inherited, see ‘‘Constructors’’ in Mail::Identity

`$obj`−>**parse**(STRING)

Parse the string for an address. You never know whether one or more addresses are specified on a line (often applications are wrong), therefore, the STRING is first parsed for as many addresses as possible and then the one is taken at random.

Attributes

Extends ‘‘Attributes’’ in Mail::Identity.

`$obj`->**address**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**charset**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**comment**([STRING])
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**description**()
Inherited, see ‘‘Attributes’’ in User::Identity::Item

`$obj`->**domain**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**language**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**location**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**name**([$newname])
Inherited, see ‘‘Attributes’’ in User::Identity::Item

`$obj`->**organization**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**phrase**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

`$obj`->**username**()
Inherited, see ‘‘Attributes’’ in Mail::Identity

Collections

Extends ‘‘Collections’’ in Mail::Identity.

`$obj`->**add**($collection, `$role`)
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**addCollection**(`$object` | <[$type], `%options`>)
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**collection**($name)
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**parent**([$parent])
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**removeCollection**($object|$name)
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**type**()
Mail::Message::Field::Address->**type**()
Inherited, see ‘‘Collections’’ in User::Identity::Item

`$obj`->**user**()
Inherited, see ‘‘Collections’’ in User::Identity::Item

Searching

Extends ‘‘Searching’’ in Mail::Identity.

`$obj`->**find**($collection, `$role`)
Inherited, see ‘‘Searching’’ in User::Identity::Item

Accessors

`$obj`−>**encoding**()

Character-set encoding, like 'q' and 'b', to be used when non-ascii characters are to be transmitted.

Access to the content

`$obj`−>**string**()

Returns an RFC compliant e−mail address, which will have character set encoding if needed. The objects are also overloaded to call this method in string context.

example:

```
print $address->string;
print $address;          # via overloading
```

DIAGNOSTICS

Error: `$object` is not a collection.

The first argument is an object, but not of a class which extends User::Identity::Collection.

Error: Cannot coerce a `$type` into a Mail::Message::Field::Address

When addresses are specified to be included in header fields, they may be coerced into Mail::Message::Field::Address objects first. What you specify is not accepted as address specification. This may be an internal error.

Error: Cannot load collection module for `$type` ($class).

Either the specified `$type` does not exist, or that module named `$class` returns compilation errors. If the type as specified in the warning is not the name of a package, you specified a nickname which was not defined. Maybe you forgot the 'require' the package which defines the nickname.

Error: Creation of a collection via `$class` failed.

The `$class` did compile, but it was not possible to create an object of that class using the options you specified.

Error: Don't know what type of collection you want to add.

If you add a collection, it must either by a collection object or a list of options which can be used to create a collection object. In the latter case, the type of collection must be specified.

Warning: No collection `$name`

The collection with `$name` does not exist and can not be created.

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