NAME

Mail::Identity − an e−mail role

INHERITANCE

```
 Mail::Identity
   is a User::Identity::Item
```

SYNOPSIS

```
 use User::Identity;
 use Mail::Identity;
 my $me   = User::Identity->new(...);
 my $addr = Mail::Identity->new(address => 'x@y');
 $me->add(email => $addr);

 # Simpler

 use User::Identity;
 my $me   = User::Identity->new(...);
 my $addr = $me->add(email => 'x@y');
 my $addr = $me->add( email => 'home'
                    , address => 'x@y');

 # Conversion
 my $ma   = Mail::Address->new(...);
 my $mi   = Mail::Identity->coerce($ma);
```

DESCRIPTION

The `Mail::Identity` object contains the description of role played by a human when sending e−mail. Most people have more than one role these days: for instance, a private and a company role with different e−mail addresses.

An `Mail::Identity` object combines an e−mail address, user description ("phrase"), a signature, pgp-key, and so on. All fields are optional, and some fields are smart. One such set of data represents one role. `Mail::Identity` is therefore the smart cousine of the Mail::Address object.

Extends "DESCRIPTION" in User::Identity::Item.

METHODS

Extends "METHODS" in User::Identity::Item.

Constructors

Extends "Constructors" in User::Identity::Item.

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

```
 -Option      --Defined in          --Default
  address                             <username@domain or name>
  charset                             <user's charset>
  comment                             <user's fullname if phrase is different>
  description   User::Identity::Item  undef
  domain                              <from email or localhost>
  language                            <from user>
  location                            <random user's location>
  name          User::Identity::Item  <phrase or user's fullName>
  organization                        <location's organization>
  parent        User::Identity::Item  undef
  pgp_key                             undef
  phrase                              <user's fullName>
  signature                           undef
  username                            <from address or user's nickname>
```

address => STRING
The e−mail address is constructed from the username/domain, but when both do not exist, the name is taken.

charset => STRING
comment => STRING
description => STRING
domain => STRING
language => STRING
location => NAME|OBJECT
The user's location which relates to this mail identity. This can be specified as location name (which will be looked-up when needed), or as User::Identity::Location object.

name => STRING
organization => STRING
Usually defined for e−mail addresses which are used by a company or other organization, but less common for personal addresses. This value will be used to fill the `Organization` header field of messages.

parent => OBJECT
pgp_key => STRING|FILENAME
phrase => STRING
signature => STRING
username => STRING

Attributes

Extends "Attributes" in User::Identity::Item.

`$obj−>`**address**()
Returns the e−mail address for this role. If none was specified, it will be constructed from the username and domain. If those are not present as well, then the **name()** is used when it contains a `@`, else the user's nickname is taken.

`$obj−>`**charset**()
Returns the character set used in comment and phrase. When set to `undef`, the strings (are already encoded to) contain only ASCII characters. This defaults to the value of the user's charset, if a user is defined.

`$obj−>`**comment**([STRING])
E−mail address −when included in message MIME headers− can contain a comment. The RFCs advice not to store useful information in these comments, but it you really want to, you can do it. The comment defaults to the user's fullname if the phrase is not the fullname and there is a user defined.

Comments will be enclosed in parenthesis when used. Parenthesis (matching) or non-matching) which are already in the string will carefully escaped when needed. You do not need to worry.

`$obj−>`**description**()
Inherited, see "Attributes" in User::Identity::Item

`$obj−>`**domain**()
The domain is the part of the e−mail address after the `@`−sign. When this is not defined, it can be deducted from the email address (see **address()**). If nothing is known, `localhost` is returned.

`$obj−>`**language**()
Returns the language which is used for the description fields of this e−mail address, which defaults to the user's language.

`$obj−>`**location**()
Returns the object which describes to which location this mail address relates. The location may be used to find the name of the organization involved, or to create a signature. If no location is specified, but a user is defined which has locations, one of those is randomly chosen.

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

`$obj`−>**organization**()
Returns the organization which relates to this e−mail identity. If not explicitly specified, it is tried to be found via the location.

`$obj`−>**phrase**()
The phrase is used in an e−mail address to explain who is sending the message. This usually is the fullname (the user’s fullname is used by default), description of your function (Webmaster), or any other text.

When an email string is produced, the phase will be quoted if needed. Quotes which are within the string will automatically be escaped, so you do no need to worry: input cannot break the outcome!

`$obj`−>**username**()
Returns the username of this e−mail address. If none is specified, first it is tried to extract it from the specified e−mail address. If there is also no username in the e−mail address, the user identity’s nickname is taken.

Collections

Extends ‘‘Collections’’ in User::Identity::Item.

`$obj`−>**add**($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::Identity−>**type**()
Inherited, see ‘‘Collections’’ in User::Identity::Item

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

Searching

Extends ‘‘Searching’’ in User::Identity::Item.

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

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 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 User-Identity distribution version 1.01, built on February 11, 2022. Website: *http://perl.overmeer.net/CPAN/*

LICENSE