

NAME

Mail::Message::Field::AddrGroup – A group of Mail::Message::Field::Address objects

INHERITANCE

```
Mail::Message::Field::AddrGroup
  is a User::Identity::Collection::Emails
  is a User::Identity::Collection
  is a User::Identity::Item
```

SYNOPSIS

```
my $g = Mail::Message::Field::AddrGroup->new(name => 'name');

my $a = Mail::Message::Field::Address->new(...);
$g->addAddress($a);

my $f = Mail::Message::Field::Addresses->new;
$f->addGroup($g);
```

DESCRIPTION

An address group collects a set of e-mail addresses (in this case they are Mail::Message::Field::Address objects).

Extends “DESCRIPTION” in User::Identity::Collection::Emails.

OVERLOADED

Extends “OVERLOADED” in User::Identity::Collection::Emails.

overload: @{}
Inherited, see “OVERLOADED” in User::Identity::Collection

overload: **stringification**
Returns the **string()** value.

overload: **stringification**
Inherited, see “OVERLOADED” in User::Identity::Collection

METHODS

Extends “METHODS” in User::Identity::Collection::Emails.

\$obj->string()
Returns the address group as string. When no name is specified, it will only be a comma separated list of addresses. With a name, the groups name will be prepended and a semi-colon appended. When no addresses were included and there is no name, then undef is returned.

Constructors

Extends “Constructors” in User::Identity::Collection::Emails.

\$obj->coerce(\$object)
Coerce an \$object into a Mail::Message::Field::AddrGroup. Currently, you can only coerce User::Identity::Collection::Emails (which is the base class for this one) into this one.

Mail::Message::Field::AddrGroup->new([\$name], %options)
Inherited, see “Constructors” in User::Identity::Collection::Emails

Attributes

Extends “Attributes” in User::Identity::Collection::Emails.

\$obj->description()
Inherited, see “Attributes” in User::Identity::Item

\$obj->itemType()
Inherited, see “Attributes” in User::Identity::Collection

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

`$obj->roles()`
 Inherited, see “Attributes” in User::Identity::Collection

Collections

Extends “Collections” in User::Identity::Collection::Emails.

`$obj->add($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::AddrGroup->type()
 Inherited, see “Collections” in User::Identity::Item

`$obj->user()`
 Inherited, see “Collections” in User::Identity::Item

Maintaining roles

Extends “Maintaining roles” in User::Identity::Collection::Emails.

`$obj->addRole($role| <[$name], %options> | ARRAY)`
 Inherited, see “Maintaining roles” in User::Identity::Collection

`$obj->removeRole($role|$name)`
 Inherited, see “Maintaining roles” in User::Identity::Collection

`$obj->renameRole(<$role|$oldname>, $newname)`
 Inherited, see “Maintaining roles” in User::Identity::Collection

`$obj->sorted()`
 Inherited, see “Maintaining roles” in User::Identity::Collection

Searching

Extends “Searching” in User::Identity::Collection::Emails.

`$obj->find($name|CODE|undef)`
 Inherited, see “Searching” in User::Identity::Collection

Addresses

`$obj->addAddress(STRING|$address|%options)`

Add one e-mail address to the list which is maintained in the group. This is a wrapper around **addRole()** adding flexibility on how addresses are specified. An `$address` can be anything which is acceptable for **Mail::Message::Field::Address::coerce()** or a list of options which will create such an object.

example: of adding an address to an address group

```
my @data = (full_name => "Myself", address => 'me@tux.org');
$group->addAddress(@data);
```

```
my $addr = Mail::Message::Field::Address->new(@data);
$group->addAddress(@data);
```

```
my $ma = Mail::Address->new(...);
$group->addAddress($ma);
```

`$obj->addresses()`

Returns all addresses defined in this group. The addresses will be ordered alphabetically to make automated testing possible: roles are stored in a hash, so have an unpredictable order by default.

example: getting all addresses from a group

```
my @addrs = $group->addresses;
my @addrs = map { $_->address } $self->roles; #same
```

Error handling

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::AddrGroup`

Error: Cannot create a `$type` to add this to my collection.

Some options are specified to create a `$type` object, which is native to this collection. However, for some reason this failed.

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: Cannot rename `$name` into `$newname`: already exists

Error: Cannot rename `$name` into `$newname`: doesn't exist

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 be 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.

Error: Wrong type of role for `$collection`: requires a `$expect` but got a `$type`

Each `$collection` groups sets of roles of one specific type (`$expect`). You cannot add objects of a different `$type`.

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/>