

---

# **gs.content.form.base Documentation**

***Release 3.0.3***

**GroupServer.org**

April 28, 2016



<b>1</b>	<b>Form style guide</b>	<b>3</b>
1.1	Basic style . . . . .	3
1.2	Terms . . . . .	4
<b>2</b>	<b>gs.content.form.base API Reference</b>	<b>9</b>
2.1	Site form . . . . .	9
2.2	Widgets . . . . .	9
<b>3</b>	<b>gs.content.form.base Content Providers</b>	<b>13</b>
3.1	Status message . . . . .	13
3.2	Widgets . . . . .	13
<b>4</b>	<b>Changelog</b>	<b>15</b>
4.1	3.0.3 (2016-03-24) . . . . .	15
4.2	3.0.2 (2015-02-26) . . . . .	15
4.3	3.0.1 (2014-10-10) . . . . .	15
4.4	3.0.0 (2014-06-11) . . . . .	15
4.5	2.2.1 (2013-08-09) . . . . .	15
4.6	2.2.0 (2013-07-17) . . . . .	16
4.7	2.1.1 (2013-05-30) . . . . .	16
4.8	2.1.0 (2013-03-20) . . . . .	16
4.9	2.0.0 (2012-07-26) . . . . .	16
4.10	1.3.0 (2011-12-05) . . . . .	16
4.11	1.2.0 (2011-06-29) . . . . .	16
4.12	1.1.0 (2010-12-22) . . . . .	16
4.13	1.0.0 (2010-10-07) . . . . .	16
<b>5</b>	<b>Indices and tables</b>	<b>17</b>
<b>6</b>	<b>Resources</b>	<b>19</b>



Contents:



---

## Form style guide

---

Almost all pages in *GroupServer* are forms, and almost all forms conform to the following *basic style* and use the same *terms*. The *GNOME Human Interface Guidelines* provided much of the inspiration for this guide.

### 1.1 Basic style

Having a consistent style makes a system easier to use. Because of this *link* to the page, its *title*, the *introduction* and the *button* used to submit the form all use similar phrasing. The widget *fields* sit between the introduction and the button. Finally, a *feedback message* is displayed.

#### 1.1.1 Link

A link to a form is always an imperative verb-phrase:

- Start a group
- Change the privacy
- Set a new password

The verb *change* is always preferred to *edit*.

#### 1.1.2 Title

The main page heading is always an `<h1>` element containing the same verb phrase as the *link*, while the `<title>` element contains the site name, group name, or profile name as appropriate.

Heading	Title
Start a group	Start a group: Example site
Change the privacy	Change the privacy: Initial group: Example site
Set a new password	Set a new password: Administrator: Example site

Initial caps, rather than title case, is used.

#### 1.1.3 Introduction

If the form has an introductory paragraph it should open with the verb in the infinitive form:

- To start a group...

- To change the privacy...
- To set your new password...

When possible, *your* is used in the introduction to convey a sense of inclusion, ownership, and responsibility.

---

**Note:** This use of *your* does lead to ambiguity about the **ownership of a group**, which is owned both by the *administrator* and each *member*. In some cases there is an over-arching concept of an *owner* (such as a corporate body) that is not reflected in this use of *your*.

---

### 1.1.4 Fields

A *noun* is used for the `title` of each field.

- Group type
- Privacy
- Password

---

**Note:** The fields are defined using *the API*, and are displayed in a consistent manner by the *Widgets* content provider.

---

### 1.1.5 Button

The button to submit the form always uses the verb from the *link* and *title*:

- Start
- Change
- Set

### 1.1.6 Feedback message

When a form has been processed a feedback message is displayed (by the *Status message* content provider). The exception is when the form *creates* something, when the created object should be shown.

The feedback message is a complete sentence that refers to the verb from the *link* in the past-tense, and clarifies what has been changed (usually linking to the object).

- You have changed the privacy for the *Initial group* to secret.
- You have set your password.

## 1.2 Terms

A consistent set of terms is used to describe the *actions* (verbs) that are carried out on *objects* (nouns).

### 1.2.1 Actions

Some actions are *general*, while many more are *specific GroupServer actions*.



## General actions

Most forms will just need to use *change* as their sole action.

**Change** Alter a group, profile, or site (see also *edit*):

Change the site About box.

**Click** Buttons and links are clicked.

- A *button* is explicitly stated as such (use “click” rather than “click on”):

Edit the Message and click the *Change* button.

- For a link just link-phrase is used:

To find more details click *view more* below.

**Edit** Alter a field (see also *change*):

Edit the Message and click the *Change* button.

**Select** Check-boxes and radio buttons are selected:

Select the privacy level from the Privacy list below. . .

**Visit** A *person* visits a page:

Visit the group page to see the current activity.

## Specific GroupServer actions

The following actions have specific meanings in GroupServer.

**Accept** A *person* will accept an invitation to join a group, or a *administrator* will accept a request to become a member. See also *decline*.

**Add** An *administrator* will add a *person* to a group, after which they become a *group member*.

**Decline** An invitation to become a *group member* issued by a *administrator* may be declined by a *person*. Conversely, a person can request to become a member, which can be declined by an administrator. See also *accept*.

**Invite** A *person* can be invited to join a group by an *administrator*. See also *add*. This is the inverse of *request*.

### Join

- The general term for a *person* becoming a *member* of a *group* (see also *add*, *invite*, *register*):

This month 10 people joined GroupServer Development.

- The specific way that a *person* that already has a *profile* becomes a *member* of a **public** *group* using the web interface (see also *register*, *subscribe*):

Join this group to add a post.

**Leave** A *person* will leave a *group* when they no longer wish to be a *member*. See also *remove*, *unsubscribe*.

**Manage** Alter the privileges and membership of a *group member*:

Manage the members of GroupServer Development.

**Register, Sign up** A *person* registers to create a *profile* and *join* a **public** *group* (see also *add* *invite*):

Register to become a member of GroupServer Development.

Since GroupServer 14.03 *register* has been preferred over *sign up*.

**Remove** A *administrator* may remove a *member* from a group. Only members are ever removed, use the term *revoke* for privileges. See also *leave*.

**Request** If a *person* wishes to join a private group then they can request membership of the group.

**Revoke** Privileges are *revoked* from a *member* to prevent the action from being confused with *remove*:

You have revoked group administrator privileges from *Example member*. They are now a normal member of this group.

**Sign in, Login** Entering an *email address* and password so GroupServer knows the *profile* to use for the subsequent actions:

Sign in to view the messages in GroupServer Team.

Since GroupServer 14.03 *sign in* has been preferred over *login*.

**Subscribe** A particular way a person becomes a *member* of a **public group** by sending a *message* to the group with *Subscribe* as the *Subject*. Prefer the term *join* in the general case.

**Unsubscribe** A particular way to leave a *group* by sending a *message* to the group with *Unsubscribe* as the *Subject*. Prefer the term *leave* in the general case.

**Verify** Prove that an *email address* is correct and working. Only an email addresses is ever verified, never a *person*.

## 1.2.2 Objects

There are three main actors in GroupServer: a *profile*, a *group*, and a *site*.

**Address, Email address** Use *address* rather than *email address*, unless there is possible ambiguity with a *Web page address* (also known as a URL). The term *email* could mean an address, or a *message* and should be avoided. Each *group* has a single address, while a *person* can have multiple addresses associated with their *profile*.

**Administrator, Group administrator, Site administrator** A person that can alter a *group* or *site*, as opposed to a *normal member*.

**Archive** The topics and posts in a *group* are known as the archive (see also *post*, and *topic*):

The group page shows the archive of posts that have been added.

**Attachment** An attachment is only used in when discussing an *email message*: they are stripped when the processed by them *group* and replaced with a link, which is then referred to as a *file*.

**File** A file is *associated* with a *post*. It appears in a list in the bottom of the post, both in the *archive* and in the *email message* sent from the group.

**Group** In GroupServer a group is analogous to a *listserv* or *forum* in other systems. A *person* is a *member* of a group, and the group belongs to a *site*. Each group has an *email address* that people can use to make a *post*, which is shown in the *archive* found on the *group page*.

**Group page** The web page for a *group*, as opposed to the *homepage*:

Visit the group page at <<http://groupserver.org/groups/development>>.

**Homepage** The web page for a *site* rather than the *group page*:

Visit the homepage at <<http://groupserver.org/>>.

**Member, Group member, Site member** A *person* that belongs to a site or group (explicitly or implicitly):

You are a member of GroupServer Development.

**Message, Email message** A message is either

- Sent to a *group* when someone makes a *post* using email, or

- Sent from the group after someone has made a post.

It can be referred to as a *message* if there is no ambiguity with the *feedback message*; the term *email* is easily confused with an *address* and should be avoided. See also *notification*.

**Normal member** A **normal** member is a group member that lacks *administrator* privileges. In a **discussion** group a normal member can add and view a *post*, while in an **announcement** group a normal member can only view posts (see *posting member*).

**Notification** An *email message* sent by GroupServer for any reason *other* than a *person* making a *post*:

Because you are a *group administrator* you will receive a notification whenever someone leaves your *group*.

**Person, User** Avoid the term *user*, preferring *person* at the very least. However, *member* or *administrator* should be used in preference to either. In all cases *they* is used as a gender-neutral singular.

**Post** A post is made to a *group* by a *member*, and organised into a *topic*. It may be posted using the web interface to GroupServer, or by sending an *email message* to the group. It may be associated with a *file*.

**Posting member** A member of an **announcement** group that can add posts to the group.

**Profile** The data associated with a person. Try and distinguish between the profile and the *person*:

To add an email address to your profile...

**Site** Each *group* belongs to a site, and a *profile* is shared between the different groups. A GroupServer site sits at the top of a domain; the index-page for this domain is referred to as the *homepage*.

**Topic** One or more posts with the same *Subject* are organised into a topic within a *group*. See also *post*.



---

## gs.content.form.base API Reference

---

The package exports the following API symbols.

### 2.1 Site form

The abstract base-class `gs.content.form.SiteForm` is a Zope PageForm provides the `siteInfo` and `loggedInUser` properties.

#### 2.1.1 Example

```
from gs.content.form.form import SiteForm

class Change(SiteForm):
    def __init__(self, context, request):
        super(Change, self).__init__(context, request)
```

### 2.2 Widgets

Four custom widgets are provided: *radio buttons*, *check boxes*, a *select*, and a *disabled text* entry.

#### 2.2.1 Radio buttons

The widget `gs.content.form.base.radio_widget()` is a variant of the standard `zope.formlib` radio button, but it has the correct association between the label and the button.

`gs.content.form.base.radio_widget(field, request)`

Create a radio-widget with a clickable label.

##### Parameters

- **field** – The field that the radio-widget is created for.
- **request** – The current HTTP request.

**Returns** A radio widget.

## Example

```
@Lazy
def form_fields(self):
    retval = form.Fields(IChange, render_context=False)
    retval['field'].custom_widget = radio_widget
```

### 2.2.2 Check Boxes

Rather than a *select*, it is often nicer to present a list of checkboxes. This is especially true if the user is supposed to select *multiple* items (which is normally done using the *Control* key in a select-box, but few people know this). The `gs.content.form.base.multi_check_box_widget()` widget displays a list of checkboxes based on a vocabulary

`gs.content.form.base.multi_check_box_widget(field, request)`  
Create a widget with multiple check-boxes.

#### Parameters

- **field** – The field that the checkbox-widget is created for.
- **request** – The current HTTP request.

**Returns** A multi-checkbox widget.

## Example

```
@Lazy
def form_fields(self):
    retval = form.Fields(IChange, render_context=False)
    retval['field'].custom_widget = multi_check_box_widget
```

### 2.2.3 Select

The standard Zope select widget annoying sets the size too small. The `gs.content.form.base.select_widget()` creates a select box that shows 15 items. If fewer items are desired then *radio buttons* or *check boxes* should be used:

`gs.content.form.base.select_widget(field, request)`  
Create a select-widget that is larger than normal

#### Parameters

- **field** – The field that the radio-widget is created for.
- **request** – The current HTTP request.

**Returns** A select widget.

**Return type** `zope.app.form.brower.SelectWidget`

## Example

```
@Lazy
def form_fields(self):
    retval = form.Fields(IChange, render_context=False)
    retval['field'].custom_widget = select_widget
```

### 2.2.4 Disabled Text

The `gs.content.form.base.disabled_text_widget` factory creates a text widget that has the CSS class set to disabled.





---

## gs.content.form.base Content Providers

---

Two content providers are used for standard parts of forms in GroupServer. One displays the *status message* at the top of the page, while the other displays the *widgets*.

### 3.1 Status message

The `groupserver.FormStatusMessage` content provider displays the status of a form (the `status` and `errors` properties of the form-view class) after it submits. It provides a standard markup for the form, and normally sits in the messages slot.

```
<tal:block
  metal:fill-slot="messages"
  content="structure provider:groupserver.FormStatusMessage"
  define="errors view/errors;
          status view/status;
          widgets view/widgets">&#160;</tal:block>
```

Two optional Boolean arguments — `showPageErrors` and `showWidgetErrors` — can be passed to the content provider to control whether page and widget errors are displayed by the content provider

```
<tal:block
  content="structure provider:groupserver.FormStatusMessage"
  define="errors view/errors;
          status view/status;
          widgets view/widgets;
          showPageErrors python:True;
          showWidgetErrors python:False"
  metal:fill-slot="messages">&#160;</tal:block>
```

`showPageErrors` defaults to `False` while `showWidgetErrors` defaults to `True`.

### 3.2 Widgets

The `groupserver.FormWidgets` content provider displays the widgets themselves. The `widgets` parameter is used to provide a list of widgets to the content provider.

```
<tal:block define="widgets view/widgets">
  <tal:block replace="structure provider:groupserver.FormWidgets"/>
</tal:block>
```

By default each widget is placed in a `<div>` that is marked as required, or not.

---

## Changelog

---

### 4.1 3.0.3 (2016-03-24)

- Adding a style guide to the documentation
- Adding some unit tests.

### 4.2 3.0.2 (2015-02-26)

- Switching the `groupserver.FormWidgets` content-provider over to a `zope.browserpage.ViewPageTemplateFile`, so internationalisation works

### 4.3 3.0.1 (2014-10-10)

- Moving to [GitHub](#) as the canonical repository
- Naming the `reStructuredText` files as such
- Adding Sphinx [documentation](#)
- Tidying the code

### 4.4 3.0.0 (2014-06-11)

- Renaming the product `gs.content.form.base` from `gs.content.form`

### 4.5 2.2.1 (2013-08-09)

- Updating the product metadata
- Making the `form-fields` property of the `SiteForm` an **abstract** property.

## 4.6 2.2.0 (2013-07-17)

- Adding options to show page or widget errors with the form-status content provider.

## 4.7 2.1.1 (2013-05-30)

- Tiding of the `SiteForm` code

## 4.8 2.1.0 (2013-03-20)

- Adding the checkbox widget
- Adding some Twitter Bootstrap support

## 4.9 2.0.0 (2012-07-26)

- Changing the import-API
- Updating the documentation

## 4.10 1.3.0 (2011-12-05)

- Adding the logged-in user as a property to the `SiteForm`

## 4.11 1.2.0 (2011-06-29)

- Adding the select-box widget

## 4.12 1.1.0 (2010-12-22)

- Adding the `SiteForm` class
- Adding the status-message content provider
- Switching from the deprecated `ustring` and `utext` types

## 4.13 1.0.0 (2010-10-07)

- Initial version.

The `gs.content.form.base` product contains many functions and classes for dealing with `zope.formlib` forms<sup>1</sup> in `GroupServer`.

---

<sup>1</sup> See <<http://docs.zope.org/zope.formlib/>>

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`



---

### Resources

---

- Documentation: <http://groupserver.readthedocs.io/projects/gscontentformbase/en/latest/>
- Code repository: <https://github.com/groupserver/gs.content.form.base>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>





## A

Accept, [5](#)  
Add, [5](#)  
Address, [6](#)  
Administrator, [6](#)  
Archive, [6](#)  
Attachment, [6](#)

## C

Change, [5](#)  
Click, [5](#)

## D

Decline, [5](#)

## E

Edit, [5](#)  
Email address, [6](#)  
Email message, [6](#)

## F

File, [6](#)

## G

Group, [6](#)  
Group administrator, [6](#)  
Group member, [6](#)  
Group page, [6](#)

## H

Homepage, [6](#)

## I

Invite, [5](#)

## J

Join, [5](#)

## L

Leave, [5](#)

Login, [6](#)

## M

Manage, [5](#)  
Member, [6](#)  
Message, [6](#)  
multi\_check\_box\_widget() (in module  
gs.content.form.base), [10](#)

## N

Normal member, [7](#)  
Notification, [7](#)

## P

Person, [7](#)  
Post, [7](#)  
Posting member, [7](#)  
Profile, [7](#)

## R

radio\_widget() (in module gs.content.form.base), [9](#)  
Register, [5](#)  
Remove, [6](#)  
Request, [6](#)  
Revoke, [6](#)

## S

Select, [5](#)  
select\_widget() (in module gs.content.form.base), [10](#)  
Sign in, [6](#)  
Sign up, [5](#)  
Site, [7](#)  
Site administrator, [6](#)  
Site member, [6](#)  
Subscribe, [6](#)

## T

Topic, [7](#)

## U

Unsubscribe, [6](#)

User, [7](#)

## V

Verify, [6](#)

Visit, [5](#)