
GroupServer Documentation

Release 15.11

groupserver.org

November 30, 2015

1	Getting GroupServer	1
1.1	Download	1
1.2	Dependencies	1
2	GroupServer installation	3
2.1	Quick start	3
2.2	Set up	4
2.3	Run the installer	6
2.4	History	8
3	Starting and stopping GroupServer	11
3.1	Introduction	11
3.2	Trying out GroupServer	11
3.3	Running GroupServer	12
3.4	Issues, and problems	13
4	Configuring postfix	15
4.1	Introduction	15
4.2	Postfix configuration	16
5	Configuring a web proxy	19
5.1	Introduction	19
5.2	Add a virtual host	19
5.3	Change the reported port	21
5.4	Change the skin	22
5.5	Secure connections: TLS, SSL, and HTTPS	22
6	Configuring cron	25
6.1	Introduction	25
6.2	Daily digest of topics	25
6.3	Monthly profile status	26
7	Connecting external systems	27
7.1	Introduction	27
7.2	Authentication token	27
7.3	Group information	28
7.4	Profile life-cycle	28
8	Translation guide	33

8.1	Introduction	33
8.2	Translate GroupServer	33
8.3	Add internationalisation (i18n)	33
8.4	Update the products	36
9	Development guide	37
9.1	Introduction	37
9.2	Development process	37
9.3	System structure	40
10	Remove GroupServer	45
11	Release notes	47
11.1	Awesome aperitifs	47
11.2	Frozen treats	76
12	Buildout	103
12.1	Installation	103
13	Resources	105

Getting GroupServer

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-03-06

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

GroupServer is distributed as a compressed tar-file that you *download*. This file contains an **installation script for Ubuntu** will install all the necessary *dependencies* that your system currently lacks (see [GroupServer installation](#)).

1.1 Download

GroupServer is distributed as a compressed tar-file. To download the latest version of GroupServer

1. Visit <<http://groupserver.org/downloads>> and click Download.
2. Extract GroupServer from the tar-file into a directory such as /opt, /home or /usr/local.

When the tar-file is extracted a new directory will be made. This directory contains the configuration files for GroupServer. In addition, the installation process will download and install some dependencies into the directory (see [GroupServer installation](#)). GroupServer will be run from the same directory.

Permissions You may need to be the superuser (`root`) to extract the archive. If you do then you **must** change the ownership of the new GroupServer directory and all of its contents to a normal user. GroupServer can only be run by users with normal privileges.

1.2 Dependencies

For your erudition, the packages that contain the programs and libraries that are required by GroupServer are listed for Ubuntu, CentOS, and RedHat Enterprise Linux. The installation script for Ubuntu runs `apt-get` to install all the requisite packages (see [GroupServer installation](#)).

	Packages	
System	Ubuntu	CentOS 6.x or RHEL 6.x
Python	python2.7	python-devel
	python2.7-dev	python-setuptools
	python-virtualenv	See <i>CentOS and RHEL</i>
GNU C++	g++	gcc-c++
Make	build-essential	make
PostgreSQL Database	postgresql	postgresql See <i>CentOS and RHEL</i>
	libpq-dev	postgresql-server
		postgresql-libs
		postgresql-devel
Postfix Email Server	postfix	postfix
	postfix-dev	
Redis	redis-server	redis
JPEG Support	libjpeg62-dev	libjpeg-devel
WebP Support	libwebp-dev	
zlib (PNG) Support	zlib1g-dev	zlib
		zlib-devel
SMTP Test	swaks	swaks

GroupServer installation

Authors Michael JasonSmith; Richard Waid; Marek Kuziel; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-08-17 (see *History*)

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

Installation can be tricky: much has to be configured and set up correctly or installation will fail. We wrote this documentation for people with moderate experience in Linux system administration. If you get stuck, please ask us questions in [GroupServer Development](#). Other more detailed guides would be gratefully accepted.

Note: GroupServer is developed on [Ubuntu](#), and is know to run on [CentOS](#). We will gladly accept any modifications you have that will make GroupServer run on more platforms.

2.1 Quick start

1. Download the latest version of GroupServer from <<http://groupserver.org/downloads>> and extract the archive.
2. Create a new hostname for your GroupServer site. Yes, you will need a new one (see *Pick a Host Name*).
3. Edit `config.cfg` (see *Configure GroupServer*).
4. Enable prepared transaction support in PostgreSQL (see *Configure PostgreSQL*).
5. Run the following (see *Run the installer*):

```
$ ./gs_install_ubuntu.sh
```

6. *Start Zope:*

```
$ ./bin/instance fg
```

7. Visit your new site.
8. Commence the *next steps*.

2.2 Set up

GroupServer builds on five major infrastructure packages. These packages are installed by the installation script. However, you must configure the relationship between GroupServer and to these packages **before** you *run the installer*.

- **Postfix** handles both the incoming and outgoing email.
- **Zope** provides the web framework and basic web server support.
- **PostgreSQL** stores the posts.
- **ZODB** stores some web content and the user-information.
- **Redis** provides an application cache.

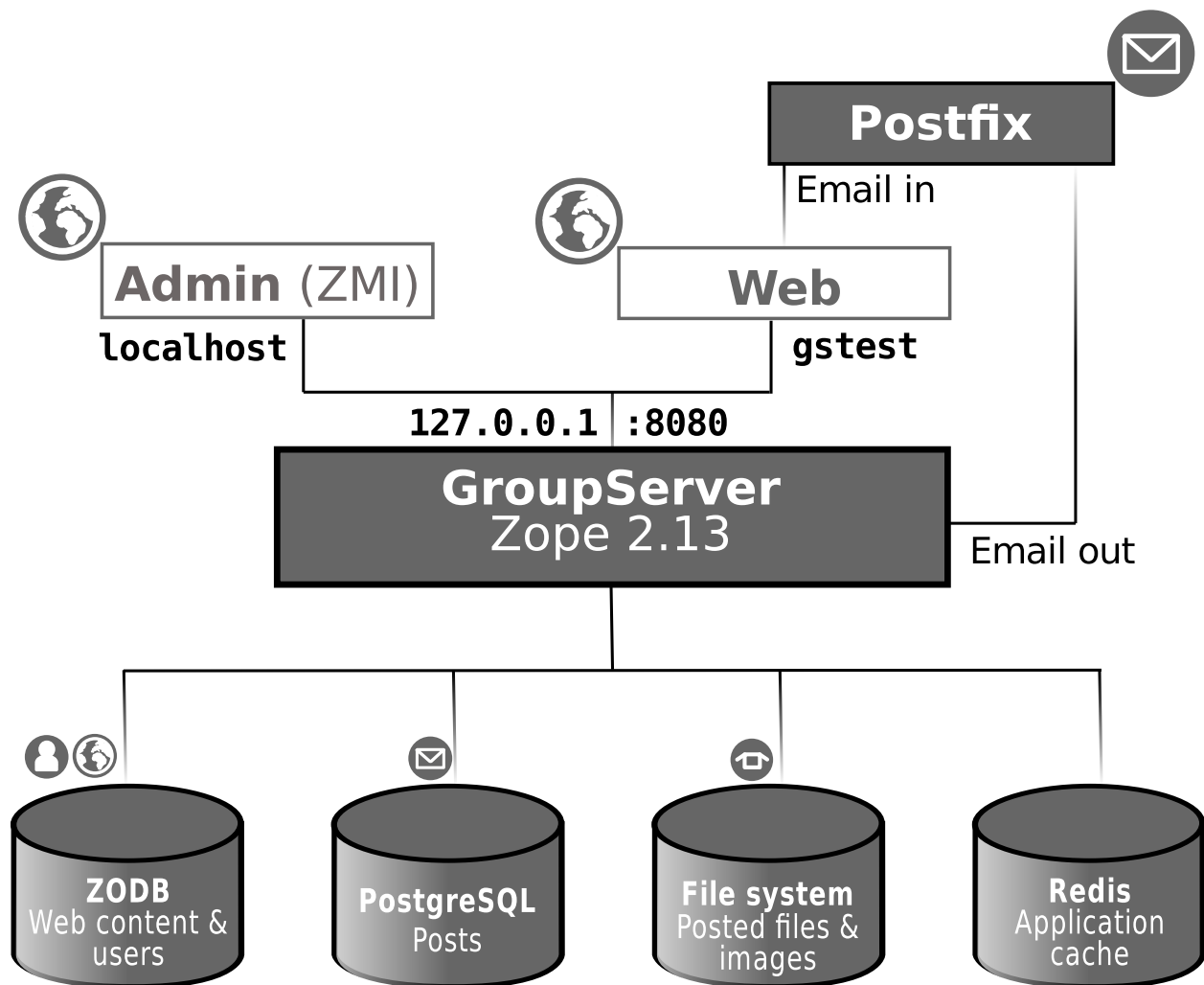


Fig. 2.1: GroupServer listens for connections on a single port (8080, by default) and serves up either the administration interface (ZMI) or the normal web interface depending on the name used to connect to the web server (virtual hosting). Email comes into the server via the web interface, and goes out using SMTP. The data is stored in a variety of locations.

Setting up GroupServer is done in four steps: first *pick a host name*, then *configure GroupServer*, *run the installer* to install the system, and finally *start Zope*.

2.2.1 Pick a host name

Your new site needs its own hostname. This is the name that people will use to access your new GroupServer site with a web browser. For a trial system, the name can be set up in the `hosts(5)` file.

1. Edit `/etc/hosts` as the `root` user.
2. Add the new host name to the `localhost` entry, which is normally the first one. For example, to add the name `gstest` change the line to the following:

127.0.0.1	localhost gstest
-----------	------------------

3. Save the `hosts` file.

2.2.2 Configure GroupServer

The configuration of GroupServer is mostly carried out by modifying the `config.cfg` file, which is located in the root of the GroupServer folder ¹. First you must configure the *GroupServer Site* itself. Next the *Zope* system, which will run your GroupServer site, needs to be configured, before the *database storage*.

GroupServer site

You will need to check all the configuration for your initial site.

host The domain name used by people accessing your new GroupServer site. It must be the same as what you picked a host name earlier (see *Pick a host name*).

admin_email When GroupServer is installed, an example site and group are created. So you can use the administration functions you must sign in as an administrator. This is the email address of that administrator. Posts to the example group will be sent to the administrator at this address. This email address **must** work.

admin_password The password of the administrator of the new GroupServer site. The password will be used to sign in, and can be changed after the site has been created.

support_email The email address where support messages are sent, and where email notifications are sent from. For testing this can be set to your own or the admin email address.

smtp_host The SMTP host that will be used to send email from GroupServer. It defaults to `localhost`, assuming you will be running Postfix on the same machine as GroupServer.

Zope

Zope is used to provide the web-framework for GroupServer, and a basic web-server. The server listens for connections on a single port (the `zope_port`) and provides the GroupServer UI if connections are made using the `host` name, or the Zope Management Interface (ZMI) if connections are made with any other host names.

The `zope_host` and `zope_port` are probably correct for most systems, whether you are testing or if you are going to proxy GroupServer (see *Configuring a web proxy*). However, for security we recommend you change the name and password of the Zope administrator.

zope_host The name of the host that will run Zope. It defaults to the local machine (`127.0.0.1`).

zope_port The IP port that Zope will listen to. It defaults to 8080, and it is recommended that you leave this value as-is, unless another service is running on port 8080. (Zope will have to run as `root` to use port 80, and this is discouraged; to use port 80 you will need to *proxy* GroupServer, see *Configuring a web proxy*.)

¹ The `cfg` files are interpreted by the standard Python `ConfigParser` module, which accepts a syntax very similar to Windows INI files.

zope_admin The name of the user who will administer Zope. This is used to log into the Zope Management Interface (ZMI).

zope_password The password for the Zope administrator. It can (and should) be changed after GroupServer has been set up.

Note: The IP-address of the `zope_host` and `host` (see *GroupServer site*) must be the same.

Database storage

GroupServer stores most of its data in **PostgreSQL**. Two passwords need to be set by you to protect this data.

pgsql_password The password required to attach to the PostgreSQL database. The install system will create a PostgreSQL database, and protect it with this `pgsql_password`.

relstorage_password The **RelStorage** system will store data in a PostgreSQL database for Zope. This data is protected by the `relstorage_password`.

2.2.3 Configure PostgreSQL

The **RelStorage** system that is used by GroupServer requires *prepared transaction* support to be enabled in PostgreSQL. To enable prepared transaction support carry out the following steps.

1. Edit the PostgreSQL configuration file. On Ubuntu you must be `root` to edit this file, which is located in `/etc/postgresql/9.x/main/postgresql.conf`. (The actual directory name may be different depending on the version of PostgreSQL you have installed; change the `9.x` to match your version as appropriate.)
2. Find the line that reads

```
max_prepared_transactions = 0
```

If the line is set to something *other* than 0 then nothing needs to change, and you can *run the installer*.

3. Change the line to read

```
max_prepared_transactions = 1
```

4. Restart PostgreSQL. On Ubuntu this is done using the following command:

```
$ sudo service postgresql restart
```

2.3 Run the installer

The installer for Ubuntu is a Bash script. (For *CentOS and RHEL* you will have to carry out the steps by hand.) To run the GroupServer installer enter the following command:

```
$ ./gs_install_ubuntu.sh
```

You will be prompted for your password. This is required to check that your Ubuntu system has met all the dependencies. Next the installer ensures that the *set up* is correct.

Permissions GroupServer can only be run by users with normal privileges. If the installation directory is owned by `root` then you must **change the ownership** of the installation directory to a normal user and switch (`su`) to that user. Then run the installer.

The rest of the installation process should be completely automatic.

- The system will create a *sandbox* for your GroupServer site, with its own version of Python, placed in `./bin/`.
- It will then configure the PostgreSQL databases that store the data for your site.
- Finally, it will start the buildout² system that will **download** and install all the requirements for GroupServer (around 47MB of packages) including:
 - [eGenix.com mx Base](#) (4.4MB)
 - [SQLAlchemy](#) (3.9MB)
 - [lxml](#) (3.4MB)
 - [Pillow](#) (2.4MB)
 - [Zope 2.13](#) (1.4MB)

You need a functioning network connection to download the GroupServer requirements.

It is a good idea to make a cup of coffee, or go to lunch, while buildout processes. The log file for the install will be written to `parts/log/{year}-{month}-{day}.{n}.log`.

2.3.1 CentOS and RHEL

The process to install GroupServer on CentOS or RedHat Enterprise Linux is manual. The basic idea is as follows, but it lacks testing.

Note: Commands that have to be run as `root` are shown on lines that begin with a `#`. Commands that must be run as a normal user are shown on lines that begin with a `$`.

1. Install the *Dependencies*.

PostgreSQL The version of PostgreSQL that is supplied with RHEL 6.x and CentOS 6.x (PostgreSQL 8.4) lacks the features required by GroupServer. You will need to install PostgreSQL 9, including the **development libraries** using [the instruction provided by the PostgreSQL project](#).

Python The version of Python supplied with RHEL 6.x and CentOS 6.x (Python 2.6) lack the features required by GroupServer. You will need to install Python 2.7 using [these instructions from H20.ai](#).

2. Create the two database users specified in `config.cfg`, using **createuser**:

```
# createuser -D -S -R -l gsadmin
# createuser -D -S -R -l gszodbadmin
```

3. Create the two databases specified in `config.cfg` using **createdb**:

```
# createdb -Ttemplate0 -O gsadmin -EUTF-8 groupserver
# createdb -Ttemplate0 -O gszodbadmin -EUTF-8 groupserverzodb
```

4. Get the Python `virtualenv` package:

```
# easy_install virtualenv
```

5. Set up a Python virtual-environment for GroupServer:

```
$ virtualenv --python=python2.7 --no-site-packages .
```

² For more on **buildout** see [Buildout](#) and [the Buildout site](#).

6. Activate the Python virtual-environment (**note** the dot-space at the start of the command):

```
$ . bin/activate
```

7. Install the `argparse` module:

```
$ easy_install argparse==1.1
```

8. Fetch the `zc.buildout` system that builds GroupServer:

```
$ easy_install zc.buildout==1.7.1
```

9. Run the buildout process:

```
$ ./bin/buildout -N
```

2.3.2 Start Zope

Your GroupServer site is supported by Zope. To start Zope run the following command:

```
$ ./bin/instance fg
```

Zope will have started when the message `Zope Ready to handle requests` is displayed in the terminal.

You should be able to view your GroupServer site at `http://{host}:{zope_port}`. If you kept the defaults, the address will be `<http://gstest:8080>`.

- The host is the one you picked earlier (see [Pick a Host Name](#)).
- The port is the one that site listens to (see [Configure GroupServer](#)).

Use `Control-c` to stop Zope.

2.3.3 Next steps

- [Starting and stopping GroupServer](#) has more information on running GroupServer, including running it as a **daemon**.
- The steps required to configure a **proxy** is documented in [Configuring a web proxy](#).
- We document the setup required to **receive email** with GroupServer in [Configuring postfix](#).
- Finally, we outline the steps required to send out the **daily digest of topics** in [Configuring cron](#).

2.4 History

Version	Date	Change
15.11	2015-08-17	Updating the Sphinx markup, and mentioning the log files
15.03	2015-03-27	Updating the CentOS install instructions.
15.03	2015-03-25	Making a note about PostgreSQL 9 on CentOS and RHEL.
15.03	2015-03-06	Moving the <i>Dependencies</i> and <i>Download</i> sections to Getting GroupServer .
14.11	2014-11-17	Renaming the <i>Requirements</i> section <i>Dependencies</i> .
14.11	2014-10-30	Moving the <i>Remove GroupServer</i> section to Remove GroupServer .
14.11	2014-10-30	Integrating updates and suggestions from Scott Fossean.

Cont

Table 2.1 – continued from previous page

Version	Date	Change
14.11	2014-10-21	Adding the setup diagram.
14.11	2014-10-14	Reducing the number of ports to one.
14.06	2014-06-23	Moving the sections for configuring the proxy and Postfix to their own documents.
14.03	2014-03-25	Clarifying the Requirements wording.
14.03	2014-03-20	Updating to Ouzo.
12.11	2012-11-27	Adding the sections <i>CentOS and RHEL</i> and <i>Configure PostgreSQL</i> .
12.11	2012-11-19	Adding a link to the Postfix documentation for Ubuntu.
12.11	2012-10-25	Removing some odd dependencies.
12.05	2012-04-30	Updating the <i>Configure GroupServer</i> and <i>Run the Installer</i> sections.
12.05	2012-04-24	Removing unnecessary dependencies, and using <code>pip</code> in the <i>Run Buildout</i> section.
11.08	2011-12-19	Adding the packages required for XML support and XSLT support on RHEL and CentOS to the list of Requirements.
11.08	2011-12-16	Adding the CentOS packages to the list of Requirements, with much thanks to Patrick Leckey .
11.08	2011-11-15	Altering the requirements to switch the <code>build-essential</code> dependency to make on the advice of David .
11.08	2011-10-27	Adding the Download section, and clarifying some more of the documentation.
11.08	2011-10-26	Correcting some mistakes, and clarifying the documentation on the advice of Ross Chesley .
11.08	2011-09-01	Reordering the subsections of <i>Configure Zope</i> .
11.07	2011-07-08	Adding the <code>build-essential</code> dependency and the cut-n-paste <code>apt-get</code> block to the Requirements.
11.06	2011-07-05	Adding the prologue.
11.06	2011-07-04	Updating the notes, because of a change to the name of the initial GroupServer instance.
11.06	2011-06-17	Added postfix configuration and spooling notes.
11.05	2011-05-26	Fixing a typing mistake, and mentioned that the <code>pgsql_dbname</code> and <code>pgsql_user</code> had to be different.
10.09	2010-09-01	Changing how the configuration options are set.
1.0 β^2	2010-07-15	Improving the buildout instructions.
1.0 β^2	2010-07-07	Changing the Zope 2.10 (Python 2.4) instructions to Zope 2.13 (Python 2.6) instructions.
1.0 β	2010-06-04	Removed a duplicated instruction from the <i>Quick Start</i> , and bumped the version number.
1.0 α	2010-05-31	Fixing a typo and adding minor improvements.
1.0 α	2010-05-01	Fixing, because upstream broke our buildout.
1.0 α	2010-04-29	Better automatic configuration, so the <i>Configure GroupServer</i> section has been removed.
1.0 α	2010-04-28	Improving the documentation for <code>gs_port</code> and added documentation for the <code>gs_admin</code> and <code>gs_user</code> options.
1.0 α	2010-04-23	Adding a link to the downloads page. Clarified the security changes that are made to PostgreSQL.
1.0 α	2010-04-06	Fixing some quoting in the requirements.
1.0 α	2010-03-31	Fixing the Requirements, adding <i>Remove GroupServer</i> and <i>History</i> .
1.0 α	2010-03-25	Fixing the config options, added <i>Quick Start</i> .
1.0 α	2009-10-04	Updating to reflect the new egg-based system.

Starting and stopping GroupServer

Authors Michael JasonSmith; Richard Waid;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-03-31

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

3.1 Introduction

In this document we present a quick introduction to starting [GroupServer](#). First you have to install GroupServer, which is covered in the `groupserver-install.txt` file in the `docs/` directory of your GroupServer folder, or [online](#). Here we will cover *trying out GroupServer*, and *running GroupServer*. Finally, we try and deal with some *issues, and problems*.

3.2 Trying out GroupServer

The first time you fire-up GroupServer it is reassuring to see all the debugging information, in case there is an error.

3.2.1 Starting GroupServer for the first time

Once GroupServer has been installed you should be able to start it with the command in the GroupServer installation directory:

```
$ ./bin/instance fg
```

This starts the [Zope](#) instance, which runs GroupServer, in *foreground* mode.

3.2.2 Access the ZMI

Generally most configuration can be done from the Web interface of GroupServer. However the *Zope Management Interface* can be used to perform some low-level tasks, and accessing it is a good indication that everything is working correctly. The ZMI is accessed by visiting `http://{zope_host}:{zope_port}/manage:`

zope_host: The name of the Zope host, that you set in the `config.cfg` file.

zope_port: The port of the Zope host, that you set in the `config.cfg` file.

If the defaults are unaltered then the URL for the ZMI will be `<http://localhost:8080/manage/>`. If all is well you will be prompted for a user-name and password. These will be the `zope_admin` and `zope_password` that you set in the `config.cfg`.

In the ZMI you should see a `groupserver` folder. Within that a `Contents` folder, leading to an `initial_site`, `groups` and finally `initial_group`.

3.2.3 Access your site

If you can *access the ZMI* without any problems then the next thing to test is if you can access your site. If you have configured DNS or your local host file correctly, you should be able to access your site at `http://${host}:${port}`:

host: The name of the GroupServer host, that you set in the `config.cfg` file.

port: The port that GroupServer is listening to, that you set in the `config.cfg` file.

If you left the defaults unaltered then the URL for your GroupServer site will be `<http://gstest:8080/>`.

Log in

Log in as an administrator by clicking the *Sign in* link and entering the `admin_email` and `admin_password` you set in the `config.cfg` file.

3.2.4 Stop the test

To stop testing type `Control-c` in the terminal where GroupServer is running.

3.3 Running GroupServer

Running GroupServer on a more permanent basis requires starting the Zope instance as a demon, and keeping a track of *the log file*.

Start the Zope instance as a demon by running the following command from the GroupServer installation directory:

```
$ ./bin/instance start
```

Stop the demon by running the following command:

```
$ ./bin/instance stop
```

3.3.1 The log file

The log file is `var/log/instance.log`, located in the GroupServer directory.

You can change the logging level, and log-file rotation, by setting parameters of the `[instance]` section of the `instance.cfg` file in the GroupServer directory. [The documentation for zope2instance](#) details the available options. Run buildout once you have made the changes:


```
$ ./bin/buildout -N
```

Then stop GroupServer and restart it.

3.4 Issues, and problems

Please, ask questions and make comments in [the GroupServer Development group](#), or in the gsdevel IRC channel on Freenode (`irc://irc.freenode.net/#gsdevel`). *The log file* will usually contain relevant information, including copies of any errors.

3.4.1 Virtual machines

With virtual machines it can be difficult to connect from your desktop — which has a Web browser — to GroupServer running on the hosted machine. The documentation for your chosen virtual environment should cover how to expose the network interface for a hosted Web service, such as GroupServer.

Configuring postfix

Authors Michael JasonSmith; Fabien Hespul

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-06-22

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

4.1 Introduction

Postfix provides the email interface for GroupServer. GroupServer uses Postfix to queue email that is delivered to groups, and pass the email messages to GroupServer.

Configuration files for Postfix that are specific to GroupServer are created when GroupServer is installed. However, it is necessary to change the main *Postfix configuration* files for the system so Postfix will send email messages on to GroupServer. The Postfix configuration will achieve the following.

- Each group will have a unique email address.
- The Postfix configuration will work, untouched, for any and all groups once it has been set up.
- A single **virtual address** will respond to all emails coming in to *any possible* group on your site.
- When an email comes in for *any* group on a it is passed on to GroupServer, which then adds the message to the correct group.

There are two configuration files, created by GroupServer during installation, that will provide what we want once they have been integrated into the existing Postfix configuration. Both are found in the `postfix_config` directory after GroupServer has been built.

groupserver.virtual: This file maps all email messages that are sent to any of your GroupServer groups to the single `groupserver-automagic@localhost` email-address. See *virtual (5)* for more information.

groupserver.aliases: This file maps the `groupserver-automagic` address to a command: piping the email to the `smtp2gs` script. See *aliases (5)* for more information.

See also:

The [Ubuntu Community Postfix Documentation](#) is useful if you are new to administering Postfix.

4.2 Postfix configuration

Below are the steps for configuring Postfix for either Debian or Ubuntu.

Note: You will need to be the `root` user to carry out most of these tasks. Commands that need to be run as root will be shown with `#` prompt, rather than a `$`.

1. Copy the configuration files from the GroupServer installation into the Postfix configuration directory.

```
# cp postfix_config/groupserver.* /etc/postfix
```

2. Change the ownership of the files to root:

```
# chown root.nogroup /etc/postfix/groupserver.*
```

If you are on a system other than Ubuntu you will need to ensure that the files are owned by the Postfix user. Running the following will display the user-name of the Postfix user.

```
$ /usr/sbin/postconf | grep default_privs | cut -f3 -d" "
```

3. Open the file `/etc/postfix/main.cf` in a text editor.
4. Update the aliases.
 - (a) Find the line that begins with `alias_maps`.
 - (b) Add the item `hash:/etc/postfix/groupserver.aliases` to the end of the `alias_maps` line. Use a comma to separate the new item from any existing items. For example

```
alias_maps = hash:/etc/aliases,hash:/etc/postfix/groupserver.aliases
```

- (c) Find the line that begins with `alias_database`.
- (d) Add the item `hash:/etc/postfix/groupserver.aliases` to the end of the `alias_database` line. Use a comma to separate the new item from any existing items. For example

```
alias_database = hash:/etc/aliases,hash:/etc/postfix/groupserver.aliases
```

5. Update the virtual alias.
 - (a) Find the line that begins with `virtual_alias_maps`. If no line exists add one after the `alias_database` line.
 - (b) Add the item `hash:/etc/postfix/groupserver.virtual` to the end of the `virtual_alias_maps` line. For example

```
virtual_alias_maps = hash:/etc/postfix/groupserver.virtual
```

6. Add the following to the bottom of the `main.cf` file, unless it is previously defined

```
smtpd_authorized_verp_clients = 127.0.0.1,localhost
```

7. Generate the Postfix hashes by running **postmap** and **postalias**:

```
# postmap /etc/postfix/groupserver.virtual
# postalias /etc/postfix/groupserver.aliases
```

8. Restart **Postfix** using **service**:

```
# service postfix restart
```

See also:

More information about the GroupServer **smtp2gs** command — including optional arguments, return values, and examples — is available from [the smtp2gs documentation](#).

Configuring a web proxy

Authors Michael JasonSmith; Fabien Hespul

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-02-12

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

5.1 Introduction

While [GroupServer](#) can run as a stand-alone web-server, it is **highly recommended** that a proxy is used when making the site available to the public to provide the following services:

- To mediate between the low level HTTP port (port 80) and the high-port that Zope runs on (normally port 8080).
- To rewrite the URL to include a skin directive.
- To rewrite the URL to support virtual hosting.
- To provide caching.
- To provide a secure connection.

In this document we explain how to *add a virtual host* to either [Apache](#) or [nginx](#), and how to *change the reported port* in GroupServer. We then explain how to *change the skin*, before we outline how to set up *secure connections*.

Note: You will need to be the root user to carry out most of these tasks. Commands that need to be run as root will be shown with # prompt, rather than a \$.

5.2 Add a virtual host

If you have a new domain ¹ that you want to use with your GroupServer installation first you must *update the GroupServer configuration* and then *add a virtual host to Apache* or *Add a virtual host to nginx*.

¹ Acquiring and configuring a new domain is out of the scope for this documentation. However, you want the A-record for your new domain to point to the IP of your GroupServer site, and the MX-record to also point at your new site.

5.2.1 Update the GroupServer configuration

If you used a host such as `gctest` to try out GroupServer then you will need to update the configuration for GroupServer itself.

1. Log into the ZMI (see [Access the ZMI](#)).
2. Visit the folder for your site at `groupserver` → `Contents` → `initial_site`.
3. Open the `DivisionConfiguration` object.
4. Set the `canonicalHost` to the domain for your new site.
5. Set the `canonicalPort` to 80.
6. Click the `Save Changes` button.

5.2.2 Add a virtual host to Apache

To add a virtual host to Apache carry out the following steps.

1. Ensure the `rewrite`, `proxy`, and `proxy_httpd` modules are enabled in Apache:

```
# a2enmod rewrite proxy proxy_httpd
# service apache2 restart
```

2. Open `/etc/apache2/sites-available/groupserver` in a text-editor.
3. Add the following to the file

```
<VirtualHost *:80>
    ServerAdmin support@example.com
    ServerName groups.example.com

    RewriteEngine on
    RewriteRule ^/(.*) http://localhost:8080/groupserver/Content/initial_site/VirtualHostBase/

    ProxyVia On

    ErrorLog ${APACHE_LOG_DIR}/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel info

    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Change the address for the site from `groups.example.com` to that of your new virtual host.
- Change the email address for `ServerAdmin` from `support@example.com` to the value of the `support_email` in the `config.cfg` file in the GroupServer directory.

4. Link the configuration for your host:

```
# cd /etc/apache2/sites-enabled/
# ln -s ../sites-available/groupserver 100-groupserver
```

5. Restart Apache using **service**


```
# service apache2 restart
```

5.2.3 Add a virtual host to nginx

Open `/etc/nginx/sites-available/groupserver` in a text-editor.

1. Add the following to the file

```
upstream gs {
    server localhost:8080;
}

server {
    listen 80;
    server_name groups.example.com;

    location / {
        rewrite /(.*) /VirtualHostBase/http/$host:80/groupserver/Content/initial_site/VirtualHost
        proxy_pass http://gs/;
        include proxy_params;
    }
}

server {
    listen 80;
    server_name zmi.groups.example.com;

    location / {
        rewrite /(.*) /VirtualHostBase/http/$host:80/VirtualHostRoot/$1 break;
        proxy_pass http://gs/;
        include proxy_params;
    }
}
```

- Change the `server_name` from `groups.example.com` to that of your new virtual host.
- Make a similar change to the second server, keeping the `zmi` at the start.

2. Link the configuration for your host:

```
# cd /etc/nginx/sites-enabled/
# ln -s ../sites-available/groupserver 100-groupserver
```

3. Reload the nginx configuration using **service**:

```
# service nginx reload
```

5.3 Change the reported port

Notifications from GroupServer (such as the *Welcome* email to a new group member) normally contain links back to the site. These links will reference the port that was used when GroupServer was built (8080) rather than the new HTTP or HTTPS port provided by the proxy. To change the port that GroupServer *says* it is using carry out the following tasks.

1. Connect to the ZMI for your site.
2. Visit the folder for your site, at `groupserver/Content/initial_site`.

3. Open the *DivisionConfiguration* object.
4. Select the check-box next to the *canonicalPort* line.
5. Click the *Delete* button. The *canonicalPort* value will be deleted.

Note: In the unlikely case that a non-standard port is used, change the value of the `canonicalPort` and click the *Save changes* button, rather than deleting the property.

5.4 Change the skin

One of the advantages of adding a proxy is it allows the skin to be easily changed. GroupServer ships with two skins: green and blue. To change the skin you must alter the rewrite rule. In the case of nginx the rewrite rule will look like the following

```
rewrite /(.*) /++skin++gs_blue/VirtualHostBase/http/$host:80/groupserver/Content/initial_site/VirtualHost
```

In the case of Apache the rewrite rule would look like the following

```
RewriteRule ^/(.*) http://localhost:8080/++skin++gs_green/groupserver/Content/initial_site/VirtualHost
```

5.5 Secure connections: TLS, SSL, and HTTPS

Setting up a secure connection is done in two stages. First you *set up your proxy*, then you *configure GroupServer*.

5.5.1 Update the proxy configuration

Establishing a secure connection is done by the proxy rather than GroupServer itself. The proxy should still listen to port 80 (HTTP) and make a permanent redirect to the secure site by returning a 301 response. In nginx the rule would look like the following:

```
server {
    listen 80;
    server_name groups.example.com;

    return 301 https://$server_name$request_uri;
}
```

The proxy will also listen to the secure port and perform a rewrite to your GroupServer site. This is similar to the rewrite when you *add a virtual host*, but

- There is configuration for the SSL certificates,
- The port is 443, rather than 80, and
- The protocol is `https` rather than `http`.

```
server {
    listen 443;
    server_name groups.example.com;

    ssl on;
    ssl_certificate /etc/nginx/ssl/groups.example.com.crt;
    ssl_certificate_key /etc/nginx/ssl/groups.example.com.key;
```

```
location / {  
    rewrite /(.*) /VirtualHostBase/https/$host:443/groupserver/Content/initial_site/VirtualHostRoot/$1;  
    proxy_pass http://gs/;  
    include proxy_params;  
}
```

You can *change the skin* in the rewrite rule, just like before.

5.5.2 Update GroupServer

GroupServer should use `https` links in email messages and in the *Share* button², to prevent potential attacks. To do this carry out the following tasks.

1. Log into the ZMI (see *Access the ZMI*).
2. Visit the folder for your site at *groupserver* → *Contents* → *initial_site*.
3. Select the *DivisionConfiguration* object.
4. Set the *canonicalPort* to 443.
5. Select the *useHTTPS* check-box (the one to the right, sorry it *is* confusing).
6. Click the *Save Changes* button.

² On the web GroupServer normally uses links without a specified protocol.

Configuring `cron`

Authors Michael JasonSmith

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-06-17

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

6.1 Introduction

The `cron` system needs to be configured to regularly execute two commands that provide some [GroupServer](#) functionality. One provides the *daily digest of topics* to those that elect to receive it. The other sends a *monthly profile status* to everyone.

See also:

The [Ubuntu Cron HOWTO](#) has more information on how to use `cron`.

6.2 Daily digest of topics

Group members can elect to receive a daily digest of topics. It summarises the activity on the group on that day, which is an excellent way to follow what is happening in a group without being overwhelmed with email.

The `cron` system needs to be configured to execute the `senddigest` command that sends out the digest:

```
00 02 * * * /opt/groupserver/bin/senddigest http://groups.example.com/
```

The `crontab(5)` example above will send the digests for all the groups on the `groups.example.com` site, at 02:00 each day. It is recommended that the digests are sent out in the early-hours of the morning, as this means they will contain the activity for the full day.

Note: The full path to the `senddigest` command in your [GroupServer](#) installation will be needed, as `PATH` is not set for `cron`.

The `senddigest` command itself normally just takes the URL of the site as its only argument. However, the [documentation for the senddigest command](#) has more information about the optional arguments, and running the command manually.

6.3 Monthly profile status

Each month GroupServer can send out a personalised email to every person that has a profile on the site and is a member of at least one group. It summarises the activity in the groups, and provides personalised suggestions about how the GroupServer experience can be improved.

The **cron** system needs to be configured to execute the **sendprofile** command that sends out the summary:

```
0 11 1-7 * * /usr/bin/test $( date +%u ) -eq 3 && \
/opt/groupserver/bin/sendprofile -t1 http://groups.example.com/
```

The `crontab(5)` example above will send the summary to everyone on the `groups.example.com` site. The command will be run at eleven in the morning of the *first Wednesday* of every month. (I recommend that you send the notification in the middle of the day in the middle of the week because it is more likely to be read.)

The [documentation for the sendprofile command](#) details the arguments. The `-t1` argument is used above to *throttle* the **sendprofile** command, slowing it down and allowing other requests to be processed. This is important because, unlike the *daily digest of topics*, the monthly profile status is run in the middle of the day when the site is likely to be busy, it is intensive to process, and is unique for every recipient.

Connecting external systems

Authors Michael JasonSmith; Eugene Guilaran;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-06-19

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

7.1 Introduction

GroupServer provides *web hooks* that allow external programs to access some features. All the hooks work the same way:

1. Data, including a *token* for authentication, is sent to a form as an HTTP POST,
2. The system processes the request, and
3. The response is returned as a JSON object.

GroupServer uses some of these hooks itself — for tasks such as for adding email (see [Configuring postfix](#)) and sending out regular notifications (see [Configuring cron](#)).

Here I first discuss the *authentication token*. I then introduce the hooks that are provided to allow external systems to retrieve *group information* and manage the *profile life-cycle*: creating, retrieving, and removing profiles.

Feel free ask any questions regarding the hooks in [GroupServer Development](#) group.

Note: The URLs used for web hooks are often quite long. This is deliberate, as it makes them easy to spot, and easier to understand as they refer to the GroupServer subsystem that provides the functionality.

7.2 Authentication token

All the web-hooks require the `token` parameter to be sent when an HTTP POST is made to the hook. The value of this token is written down in the `etc/gsconfig.ini` file, as the `token` parameter for the `webservice` section of the configuration.

A unique token is created when you install GroupServer. It is *very important* that you keep this token **secret**. If the token accessed by a nefarious individual then they can do **extensive harm** your site.

However, all is not lost *if the token is exposed*, as you can generate a new token.

Warning: Because consequence of the token being divulged is so high it is recommended that the hooks are only used from the same machine as GroupServer (so it never travels over a network), or using **secure connections** (TLS).

7.2.1 If the token is exposed

Generate a new token if the token used to authenticate the web hooks is exposed.

1. Run the **gs_auth_token_create** program, in the `bin` directory of your GroupServer install.
 - It takes a data-source name (DSN) as its only argument, identifying the **PostgreSQL** database to connect to.
 - The DSN for your GroupServer site is listed as the `dsn` parameter for the database section of the configuration in the `etc/gsconfig.ini` file.
2. The **gs_auth_token_create** program will
 - **Generate** a new token,
 - **Change** the value of the token in the PostgreSQL database, and
 - **Report** the value of the token.

At this point the token in the PostgreSQL database and the `etc/gsconfig.ini` file will be different, so *all web hooks* will be **broken**. This includes the hook that used by the **smtp2gs** program, which adds email messages to the site.

3. Edit the `etc/gsconfig.ini` file and **change** the `token` parameter in the `webservice` section. **Save** the file. All web hooks will be running again. No restart of GroupServer is necessary to change the token.

See also:

The documentation at [Read the Docs](#) contains more details about the **gs_auth_token_create** program.

7.3 Group information

The web hook `/gs-group-groups.json` is the simplest web-hook. It takes the *authentication token* (token) and the action (`get`) — and it returns a list of group-objects.

See also:

The documentation for the **Groups web-hook** has more details about how the hook works, including examples.

7.4 Profile life-cycle

There are web-hooks for managing the entire life-cycle of a profile.

- Create a profile when you *add a profile* to a group for the first time,
- Find more about people when you *retrieve profile information*.
- Finally, you can *remove a profile* from a group (or site).

Most of the profile-related web hooks return the same *profile data*.

7.4.1 Profile data

The profile data returned by the hooks involved in the *profile life-cycle* all return the same properties for the profiles, either as a single JSON object, as part of a list, or as a property of another object.

class ProfileData ()

The profile-data includes the following five properties.

id

The identifier of the profile.

name

The name of the person.

url

The URL of the profile.

groups

A list of identifiers for the groups that the person is a member of.

email

The email addresses associated with the profile.

all

A list of all the email addresses.

preferred

A list of the preferred address or addresses.

other

A list of verified addresses that are not preferred.

unverified

A list of the unverified addresses.

Example profile data

In the example JSON object below is the profile for someone called A Person. They have set a nickname, so the URL to the profile does not contain their profile-identifier. They have two email addresses — with their home address preferred, and no unverified addresses. Finally, the person belongs to two groups: Example, and Test.

```
{
  "id": "qK7SgjsTHcLNrJ2ClevcJ0",
  "name": "A Person",
  "url": "https://groups.example.com/p/aperson",
  "email": {
    "all": [
      "a.person@home.example.com",
      "a.person@work.example.com"
    ],
    "preferred": [
      "a.person@home.example.com"
    ],
    "other": [
      "a.person@work.example.com"
    ],
    "unverified": []
  },
  "groups": [
    "example",
    "test"
  ]
}
```

```
    "test"  
  ]  
}
```

7.4.2 Add a profile

The web-hook `/gs-group-member-add.json` is used to add a profile to a group. It will also create a profile, if one does not exist for that person already. The hook takes

- The *authentication token* (token),
- A name (fn),
- an email address (email),
- A group identifier (groupId), and
- An action (add).

It returns the *profile data* of the person that has been added to the group, as well as some details about whether a profile was created, or already existed.

See also:

The [documentation for the Add a profile web-hook](#) has more details about how the hook works, including examples.

7.4.3 Retrieve profile information

There are three ways to retrieve profile information: information about *an individual*, and information about *people that belong to a site*.

Single profile

The web-hook `/gs-search-people.json` allows you to retrieve information about an individual, using their user-identifier or email address. The hook takes

- An *authentication token*,
- The identifying information about someone (user) — which is either the user-identifier or email address), and
- An action (search).

It returns the *profile data* of the person, or an empty object (`{ }`) if no one could be found.

See also:

The [documentation for the Search for people web-hook](#) has more details about how the hook works, including examples.

Site members

The web-hook `/gs-site-member.json` allows you to retrieve information about the site members in a couple of ways.

- If passed an *authentication token* and an action of `users` then a simple list of user-identifiers is returned.
- If passed an *authentication token* and an action of `user_groups` then the full *profile data* is returned for each person.

See also:

The documentation for the Site members [web-hook](#) has more details about how the hook works, including examples.

7.4.4 Remove a profile

The web-hook `/gs-group-member-leave.json` removes a person from a group. The hook takes

- The *authentication token* (`token`),
- A group identifier (`groupId`), and
- A user-identifier (`userId`).

See also:

The documentation for the [Leave group web-hook](#) has more details about how the hook works, including examples.

If you only have an email-address for the person, then you should retrieve a *single profile* first to determine the user identifier (*id*).

The profile is also useful for removing someone from a **site**. A person is removed from a site when they are removed from all groups on the site: so by iterating through the list of groups (*groups*) you will eventually remove someone from the site.

Translation guide

Authors [Alice Rose](#); [Michael JasonSmith](#);

Contact [Michael JasonSmith](#) <mpj17@onlinegroups.net>

Date 2015-02-18

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

8.1 Introduction

[GroupServer](#) is written in English, and the interface has been partly translated into French and German. In this guide we work through how to *translate GroupServer*, how to *add internationalisation (i18n)*, and finally we discuss how to *update the products*.

8.1.1 Change the language

Your browser determines the language that is chosen when GroupServer displays a page. To change what language is chosen alter the *Preferred languages*, or *Language and input* setting in your browser preferences (options).

8.2 Translate GroupServer

Anyone can help improving the translations of GroupServer! We use the [Transifex system](#) to help make the translations: it is a web-based system that allows you to translate GroupServer bit by bit. **All you need is a browser.** If you have any questions please feel free to ask away in the [GroupServer development group](#).

8.3 Add internationalisation (i18n)

Adding internationalisation support to a product that lacks it is a development task. If you come across a component of GroupServer that lacks a translation please **ask for i18n to be added** in the [GroupServer development group](#). The person who responds (probably either [Michael JasonSmith](#) or [Alice Rose](#)) will then carry out the following tasks.

1. Identify the product. (The element identifiers in the HTML often point to the product that needs to be changed.)

2. Add a `locales` directory to the product, in the same directory that has the `configure.zcml` file.
3. Add `i18n` to the Python code.

- (a) To the `__init__.py` for the product instantiate a message factory, passing the name of the product as an argument:

```
from zope.i18nmessageid import MessageFactory
GSMessageFactory = MessageFactory('gs.groups')
```

- (b) Identify the Python products that contain strings that need translating. To each add the following import:

```
from . import GSMessageFactory as _
```

- (c) Add `i18n` to all the strings:

- All strings, including the simple ones, get a label with the default (English) text following. The label make Transifex much easier to deal with.

```
@form.action(name="change", label=_('change-action',
                                     'Change'),
              failure='handle_change_action_failure')
def handle_invite(self, action, data):
```

When actually adding `i18n` to a command button in a `zope.formlib` form remember to **set a name**, that way the element-identifier will be the same no matter the language.

- Complex strings have a mapping keyword argument, and a `${ }` template syntax (rather than `{ }` for Python format-strings).

```
_('start-status',
  'The group ${groupName} has been started.',
  mapping={'groupName': r})
```

4. Add `i18n` to the page templates.

- (a) Add the `i18n` namespace to the page template, using the product name as the domain.

```
<html xmlns:i18n="http://xml.zope.org/namespaces/i18n"
      i18n:domain="gs.group.start">
```

- (b) Add `i18n:translate` attributes to all elements that require translation. Always set the translation ID.

```
<p id="group-id-error" style="display:none;" class="alert"
  i18n:translate="group-id-used">
  <strong class="label alert-label">Group ID In Use:</strong>
  The Group ID <code>above</code> is already being used.
  Please pick another group ID.
</p><!--group-id-error-->
```

- (c) Add `i18n:name` attributes to dynamic content.

```
<span class="group" i18n:name="groupName"
      tal:content="view/groupInfo/name">this group</span>
```

- (d) Add `i18n:attributes` attributes to dynamic attributes.

```
<a title="Change this About box"
  i18n:attributes="title admin-change-button-title">Change</a>
```

5. Add i18n to the Zope Configuration file.

(a) Add the i18n namespace

```
<configure xmlns="http://namespaces.zope.org/zope"
  xmlns:browser="http://namespaces.zope.org/browser"
  xmlns:i18n="http://namespaces.zope.org/i18n">
```

(b) Add the registerTranslations element

```
<i18n:registerTranslations directory="locales" />
```

6. Run the latest version of i18n.sh¹ in the base directory of the product to create and update the translation.

7. Fill out the *English* translation, which is used as the source translation for Transifex.

8. Commit the changes.

9. Add the product to Transifex².

(a) In the [GroupServer organisation on Transifex](#), click on *Add project*.

(b) Fill in the Project Details form:

- Use the GroupServer product identifier as the name (e.g. gs.site.about).
- Source language is always English.
- The License is always “Permissive open-source”
- Source Code URL is the GitHub URL.

(c) Assign to the project to the GroupServer Team.

(d) Skip “Add content”.

(e) Create the project.

(f) View the new project.

(g) Choose the *Manage* link.

(h) Under *Project URL*, add hyphens where Transifex has removed dots from the project name (e.g. gssiteabout → gs-site-about).

(i) Optionally add a *Long Description* from the *Introduction* section of the product README.rst.

(j) *Save*.

(k) Update the README.rst to include a Transifex link in the *Resources* section.

```
- Translations:
  https://www.transifex.com/projects/p/gs-group-encouragement/
```

(l) Initialise the product, accepting the defaults:

```
$ tx init
```

¹ Download i18n.sh from <http://groupserver.org/downloads/i18n.sh>. It wraps the marvellous i18ndude: \$ pip install i18ndude

² Ensure you have transifex-client 0.11.1 beta or later installed: \$ pip install transifex-client==0.11.1.beta

(m) Run `tx-set.sh`³ in the base directory of the product.

(n) Sync local source and translations to remote:

```
$ tx push -s -t
```

(o) Pull the translations, now modified by Transifex from remote to local:

```
$ tx pull -a
```

(p) Commit the Transifex configuration (`.tx/`) and the modified translations to version control.

10. Push all the changes to the repositories.

8.4 Update the products

Transifex and the product need to be kept in sync with each other. When the product changes it is necessary to *update Transifex with the new strings*. Likewise, when some translations have been completed it is necessary to *update the product with the new translations*.

8.4.1 Update Transifex with the new strings

To update a Transifex project with the new strings in a product carry out the following tasks.

1. Update the `pot` file and the `po` files by running the `il8n.sh` script in the root of the product¹.
2. Update the *English* `po` file, copying the default text into the `msgstr`. This is the *source* language that supplies the example text in Transifex. (Without this step the translations can still take place, but the translators see the message identifiers, rather than the default text.)
3. Push the changes in the source file to Transifex, using the Transifex client (`tx`):

```
$ tx push -s
```

4. Commit and push the changes to the source-code repositories.

8.4.2 Update the product with the new translations

To update a product with the new translations in a Transifex project carry out the following tasks.

1. Pull the updated translations (in `po` files) from the Transifex project using the Transifex client (`tx`):

```
$ tx pull -a
```

2. Ensure that Zope is set up to automatically compile the `po` files to `mo` files:

```
$ export zope_il8n_compile_mo_files=true
```

3. Start your development system. *Change the language* in your browser to test the different translations.

Note: Browsing the Web with a changed language will result in Goggle, Microsoft, the NSA, and Yahoo! getting some funny ideas about that languages you can comprehend.

4. Commit and push the changes to the source-code repositories.

³ Download `tx-set.sh` from <<http://groupserver.org/downloads/tx-set.sh>>

Development guide

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-02-18

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

9.1 Introduction

In this guide I hope to introduce you to the tools used in the *development process* for GroupServer. I also hope to provide an overview of the *system structure*.

You can find out more about GroupServer development by reading **the archive**, and asking questions, in the GroupServer development group, joining the #gsdevel channel on Freenode.net, or both.

9.2 Development process

The development process for GroupServer is straight forward, but it may be different to what you are used to simply because of the size and structure of the project. Here we cover using a *virtual environment*, before introducing you to the *version control system* used for GroupServer.

9.2.1 Virtual environment

The development for GroupServer takes place within the Python *virtual environment* of a GroupServer installation (see [GroupServer installation](#)). When you start development you need to **activate** the virtual environment:

```
$ ./bin/activate
```

The command-prompt will change to indicate that you are now in a virtual environment.

When you have done with development you can **deactivate** the environment:

```
$ deactivate
```

9.2.2 Version control system

Two version control systems are used to store the source-code for GroupServer:

Git: [GitHub](https://github.com/groupserver) hosts the canonical repository at [<https://github.com/groupserver>](https://github.com/groupserver)

Mercurial: The Mercurial repository is hosted on a Redmine site that has a self-signed HTTPS certificate [<https://source.iopen.net/groupserver>](https://source.iopen.net/groupserver)

You can use either system, but I recommend that people use GitHub because they are more likely to have an account there. However, you will almost certainly lack the permissions required to *push* any changes. To ensure your changes are not lost you should [fork a repo](#) and commit any changes you make there. Once you are done you can make a [pull request](#).

9.2.3 Mr Developer

I recommend that you use [mr.developer](#) to integrate your development with the [Buildout](#) system that builds GroupServer. Its *usage* is simple (which is why we use `mr.developer`) and its behaviour is controlled by the *development configuration file*.

Usage

Run the following To use `mr.developer` to **checkout** the code from the VCS, and active the source code so you can work on it:

```
$ develop checkout gs.group.home
```

This will checkout the `gs.group.home` product from its [repository](#) into the `gs.group.home` directory within the `src` directory of your GroupServer installation. (There is more on [products](#) below.)

Next you will have to rebuild GroupServer to update all the configuration to point to the new code:

```
$ buildout -N
```

Now any changes that you make to the `gs.group.home` product will change your version of GroupServer.

When you have finished making changes you want you should commit them, and push your changes up to a repository.

To resume using the standard version of a product **deactivate** the source code version of the product and rebuild GroupServer:

```
$ develop deactivate gs.group.home
$ buildout -N
```

Development configuration file

The configuration for `mr.developer` is in the `develop.cfg` file, which is a configuration file that is very similar to the configuration files that control the rest of the build.

The main configuration is in the `[sources]` section. This maps each product that makes up GroupServer to the repository location. Each line is of the form

```
name = kind url
```

name: The name of the product.

kind: The kind of version control system to use. For GroupServer development this will be either `hg` or `git`. For your own development you may want to live dangerously and use `fs` for a product that lacks version control ¹.

url: The location of the repository. This is specific to the version control system.

When you make **your own changes** you will need to change the URL to point to the repository provided by the *version control system* that you use. The *default configuration* that ships with GroupServer points to the canonical Git repositories for all products.

Default configuration

The default configuration for `mr.developer` is generated from the `versions.cfg` file using the following **awk** script. It specifies that **git** should be used with all the products.

```
BEGIN {
    FS=" = "
    vcs="git"
    dest="ssh://git@github.com:/groupserver/"
    print "[buildout]"
    print "extensions = mr.developer"
    print "sources = sources"
    print "auto-checkout = "
    print "\n[sources]"
}
$1 ~ /^(gs)|(Products.G)|(Products.XWF)).*/ {
    printf "%s = %s %s\n", $1, vcs, dest, $1
}
```

To change Mr Developer to use **Mercurial** as the default VCS, but use GitHub as the primary repository, carry out the following tasks.

1. Install the **Hg-Git** plugin for Mercurial.
2. Copy the above **awk** script to the file `emit-devel.awk`.
3. Change the `vcs` variable to `hg`.
4. Add `git+` to the start of the value for the `dest` variable.
5. Run the command:

```
$ awk -f emit-devel.awk < versions.cfg > new-develop.cfg
```

6. Check that the new configuration is to your liking and move the new configuration into place:

```
$ mv new-develop.cfg develop.cfg
```

9.2.4 Adding a new product

GroupServer is split into multiple *products*, each controlling one aspect of the system. (There is more on *products* below.) To add your own functionality, or override existing functionality, you will need to add your own product.

To add your own new product to GroupServer carry out the following tasks.

1. Create the product in the `src` directory. Normally I copy an existing product:
 - Rename the product, the directories in the product namespace, and the configuration in the `setup.py`.

¹ I recommend that you use a local Mercurial repository on your local machine, rather than abandoning version control altogether.

- Delete the old code — keeping a blank `__init__.py` at the top, and the `__init__.py` files needed for the *namespace*.
 - Delete the contents of the `<configure>` element of the `configure.zcml` file, keeping the element itself.
2. Add the name of your product to the `custom-zope-eggs` section of the `custom.cfg` file.
 3. Add the version-control information for the product to the *development configuration file*.
 4. Activate the product:

```
$ develop activate your.product.name
```

5. Rebuild GroupServer:

```
$ buildout -N
```

9.3 System structure

GroupServer belongs to a family of systems that share underlying technology:

- Zope component architecture
- Plone
- Pyramid
- Chameleon
- Python

The source-code for GroupServer is split into many *products*, with the *documentation* provided by `reStructuredText`.

9.3.1 Products

GroupServer is split into many (currently 145) *products*: small Python packages that deal with one aspect of the system. The general rule is that **one product for each user interface** (usually a form). While this may seem limiting, each product usually contains

- The *page templates* that makes up the interface,
- The JavaScript that is specific to the page,
- The Python code that defines the behaviour of the interface,
- The Python code that handles storing the data and retrieving the data (using `SQLAlchemy`),
- The SQL code that defines any product-specific tables,
- The user-help, and
- The code documentation.

This tends to be more than enough for each product.

If more than one product relies on the same code then that code is normally refactored into a **base product** — which is normally given a name ending in `.base`, such as `gs.group.list.base`.

Development is carried out on one, or a few, products at a time. If you are unsure what products provide aspect of GroupServer it would be best to **ask in ‘GroupServer development’** or in `#gsdevel` on IRC. However, there are some clues: normally name of the product will make up the part of the identifier or class-name of an element in

the HTML source of a page. For example, the link to the ATOM feed of posts on the *Group* page has the identifier `gs-group-messages-posts-link` — which indicates that it is provided by the `gs.group.messages.posts` product.

```
<link id="gs-group-messages-posts-link" rel="alternate"
      type="application/atom+xml"
      title="Posts in GroupServer development"
      href="/s/search.atom?g=development&t=0&p=1" />
```

Each product makes use of *namespaces*, and *ZCML*. Each product usually contains some *static resources*, *page templates*, and some *documentation*.

Namespaces

The products use *namespace packages* (PEP 420).

- Each GroupServer product belongs beneath the `gs` namespace. Beneath that there are many others. For example, `gs.group` products deal with groups, while `gs.profile` products deal with people. Each part of the namespace is separated by dots. For example, the code that produces for the plain-text version of an email message is `gs.group.list.email.text`.
- The root of each product contains the packaging information for that product, particularly in the `setup.py` file.
- The Python code is within nested sub-directories beneath the product directory, such as `gs/group/list/email/text`. All but the last directory will have `__init__.py` files that set the directory up as a *namespace directory*. The last directory (`text` in this example) will have an `__init__.py` that may be blank. It is necessary to turn the directory into a Python module.

The Python code is made up of an `__init__.py` that is often blank, with each class in its own file. (This is my habit, you do not have to follow it.) To determine the relationship between the files, and the rest of GroupServer, it is necessary to look at the *ZCML* file.

ZCML

The Zope Configuration Markup Language (ZCML) defines the *static resources*, the *page templates*, the relationship that the Python files have to each other, and to other products. The configuration for each product is always called `configure.zcml`, and it is always in the same directory as all the Python files.

To begin with the three most important directives are as follows.

<browser:resource />: A *static resource*.

<browser:page />: A page on the web, pointing to a *page template*.

<browser:viewlet />: Part of a page, which also points at a *page template*.

Static resources

Static resources are simply files with names, which are useful for JavaScript, CSS, and images. When requested Zope sends the static file to the browser.

The resource is defined by the `<browser:resource/>` directive in the *ZCML*.

```
<browser:resource
  name="gs-group-messages-topic-compose-20140327.js"
  file="browser/javascript/compose.js"
  permission="zope2.Public" />
```

- The `name` attribute is the of the resource. The URL is made up of `/` and the name. Normally the name of the product (`gs.group.messages.topic` in this case) makes up part of the name to prevent namespace clashes, and so it is easier to work back from the filename to the product. The name *should* end with the date the resource was created so there are fewer caching issues when the resource is updated.
- The `file` is the static file that is served. It is a path from the directory that holds the *ZCML* file. Resources are always within the `browser` sub-directory, within a `javascript`, `images` or `css` directory.
- The `permission` is the permission on the resource. It is **always** `zope2.Public`. This will allow the resource to be cached.

In GroupServer the resources are always accessed from the root of the site, with `++resource++` added to the start of the name: `<http://groupserver.org/++resource++gs-group-messages-topic-compose-20140327.js>`

Page Templates

Pages themselves are defined by one of two directives in the *ZCML*: `<browser:page/>` and `<browser:viewlet/>`. The former links the Python code (`class`) with a template, giving it a name.

```
<browser:page
  name="index.html"
  for="gs.group.base.interfaces.IGSGroupMarker"
  class="gs.group.base.page.GroupPage"
  template="browser/templates/homepage.pt"
  permission="zope2.View" />
```

A viewlet is **part** of a page. It also links a `class` up with a name and template.

```
<browser:viewlet
  name="gs-group-message-topic-summary-stats"
  manager=".interfaces.ITopicSummary"
  template="browser/templates/summarystats.pt"
  class=".summarystats.SummaryStats"
  permission="zope2.View"
  weight="0"
  title="Topic Summary Statistics" />
```

The pages are created using *Zope Page Templates* (ZPT), which is the same template system that Plone uses, and is very similar to *Chameleon*.

- The page templates are always stored in a directory called `browser/templates`, within each product. Each has the extension `.pt`.
- The template itself is **XHTML 5**: the XML form of HTML 5.
- The *dynamic* parts of the template are defined by **attributes**, using the Template Attribute Language (TAL). This accesses attributes and methods of the Python code. In the following example the group-name is written into the `<h1/>` element by the `tal:content` attribute.

```
<h1 id="gs-group-home-h" class="fn"
  tal:content="view/groupInfo/name">Group</h1>
```

- Within each attribute is one or more expressions that generates the text that is placed into the page. The Python code (the `class` in the *ZCML* above) is always referred to as `view`, and a `/` is used as an attribute separator (rather than `.` in Python code). In the above example the Python class (`gs.group.base.page.GroupPage`) is accessed to get the group-information attribute (`groupInfo`), and from that the group-name is retrieved.

9.3.2 Documentation

The development documentation for [GroupServer](#) is entirely in [reStructuredText](#), with the [autodoc](#) plugin for [Sphinx](#) used to generate the source-code documentation where possible. The documentation is then pushed up to the [the GroupServer project at Read The Docs](#).

Remove GroupServer

Authors Michael JasonSmith

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2014-10-30

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

To remove [GroupServer](#), you must remove the database, the associated database user, and the directory that contains the GroupServer install.

1. To remove the database, run the following commands:

```
$ dropdb gtestdb -U postgres
$ dropuser gtest -U postgres
```

gtestdb The name of the test database.

postgres The name of the admin of PostgreSQL.

gtest The name of the PostgreSQL user.

2. Remove the directory that contains the GroupServer install:

```
$ rm -r groupserver-14.03
```

Release notes

GroupServer uses a year-month version number (yy.mm). What would be termed Version 1 with semantic version numbers, the *Frozen treats* series, corresponds to the releases from GroupServer 10.09 — Spaghettieis with a Wafer of Confusion up to and including GroupServer 12.05 — Faloodeh Consumed with an Eye on History ¹.

Currently, GroupServer could be considered in Version 2 since the release of GroupServer 12.11 — Absinthe acquired arbitrarily, which marked the start of the *Awesome aperitifs* series.

11.1 Awesome aperitifs

11.1.1 GroupServer 15.11 — Limoncello to ward off summer

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-11-02

Organization GroupServer.org

Copyright This document is licensed under a Creative Commons Attribution-Share Alike 4.0 International License by OnlineGroups.net.

Introduction

The major *changes to GroupServer* in the Limoncello release include HTML formatting of posts, better notifications, and more ways to connect with other systems. You can *get Limoncello* immediately.

Changes to GroupServer

The most visible change to GroupServer in this release is *HTML formatting of posts*. Limoncello also includes an *updated topic digest*, a new *profile status notification*, the introduction of *restricted groups*, *updated member management*, *easier sharing of links*, new *web hooks*, more *code documentation*, and an improved *German translation of GroupServer*. Finally, there have been some *minor improvements*.

¹ The project itself dates from 2007, with versions prior to 10.09 being part of the Version 1-alpha series. The first release made in May 2008. Release notes were not made until GroupServer 10.09 — Spaghettieis with a Wafer of Confusion.

HTML formatting of posts

The formatting of posts is the largest change in the Limoncello release of GroupServer. Both *post formatting in email* and *post formatting on the web* has been changed.

Post formatting in email GroupServer now will format email messages from a group using HTML — in addition to sending a plain-text version of the message.

The HTML form of the message includes additional **metadata** about the post:

- The profile image of the person who made the post
- The name of the topic that the post belongs to
- The name of the person who made the post, and a link to their profile
- The full name of the group, and a link to the group
- Links that make it easy to
 - Reply to the post
 - Start a new topic
 - View the topic on the web
 - Leave the group
 - Switch to a daily digest of topics

The HTML version of the **message body** is generated from the plain text version of the post. The formatting follows what is used on the web, which is discussed further in *the next section*.

Note: We hope to generate the HTML form of an email from the submitted HTML in a future version of GroupServer. This task is tracked in [Feature 3676](#).

Sending a HTML formatted message based on a plain-text email closes [Feature 4160](#).

Post formatting on the web The formatting of the **message body** the web has been changed to follow the updates in *post formatting in email*.

- Text in *asterisks* is made bold (as with prior releases).
- Quoted text (`> like this`) is now muted.
- URLs and site names are made into links, as before.
 - The host name in links is now made bold, so people can more easily see where the link goes.
 - Long URLs are now shown in a small-font, so they take up less room.
- Email addresses are made clickable.
 - All email addresses are shown in the body of the email messages that are sent from the group.
 - All email addresses are still shown on the web in private, restricted, or secret groups.
 - As before, addresses shown on the web in **public groups** are hidden. However, if the address is either the **group email** address or the **support address** for the site then the address is now shown.
- Posts on the web still embed **videos** from YouTube and Vimeo. In this release the videos are now embedded using `<iframe>` elements. Embedding cannot be done in the email messages sent from the group, so the videos are kept as URLs.

- **Bottom quoting** and the signature is still hidden. However, the system has been improved in the Limoncello release.
 - The *closing* (such as “Yours sincerely...” or “Kind regards...”) is now always shown in messages.
 - Extraneous CSS added to the bottom-quoting by Mozilla Thunderbird is now hidden by default.
 - The **Rest of post** section of a message on the *Post* page is now open by default. This makes following the link from the email message to the archive easier.

Attachments are still stripped, and replaced with links to the files on the web. The file-sizes of attached files now look better; empty files are explicitly labelled as `empty`.

In addition to the user-visible changes, the code for that displays the **posts on the web** has been [refactored and documented](#).

Updated topic digest

Group members can opt to receive a *daily topic digest* from a group — rather than getting an email message every time someone posts. The Limoncello release includes significant updates to the look of the digest, and the tools for sending the digests.

- The digests look like standard email notification, closing [Feature 3985](#).
- The digest now includes a photo of the most recent person to post to each topic.
- It is now easier to
 - Find the group from the digest,
 - Post to a new topic, and
 - Change your email settings.

The **senddigest** command, which is run once a day by **cron** to send the digests, has also been updated. It is now faster, and now has a `--verbose` option for producing verbose output, including a percentage-progress indicator.

Because the new *profile status notification* reminds people that they are in groups the **weekly** topic digest, which was sent when there was no activity in a group for a week, is no longer sent.

Profile status notification

GroupServer now has the ability to send out a notification that reminds people about their profile status, what groups they are in, and encourages the group members to enhance their profiles. The new *What is going on in your groups* notification is designed to be sent out once a month (towards the start of every month). The system includes a new **sendprofile** command — which works much like the **senddigest** command that sends out the daily digest of topics (see [Configuring cron](#) for more information).

There are also two new email-commands: `Status off` and `Status on`. The former records that the person wishes to stop receiving the monthly summary, the latter turns it on. Both work for a *support* group.

The creation of a profile status notification closes [Feature 370](#).

Restricted groups

A new privacy level has been added to GroupServer in the Limoncello release: *restricted groups*. Everyone that is a member of the **site** can see a restricted group, and the posts within it. It joins the three existing privacy levels:

- Public, where the group and posts are shown to everyone,

- Private, where only group-members can see the posts, and
- Secret, where only members can see the group and posts.

The different privacy levels can be set from the *Change privacy* page, linked from the *Admin* area of the group page.

Allowing the restricted group-type to be set closes [Feature 4169](#).

Configurable *Reply-to*

The *Reply-to* header for posts sent from a group can now be easily configured — using the *Reply to* property on the *General group properties* page, which is linked from the *Admin* section of the group page. In the Rakı release of we added the ability for GroupServer to change the *Reply-to* header to the email address of author of the post, the group, or both (see [Email processing](#)). However, there was never an easy way to change what the value should be. Adding this ability closes [Feature 4051](#).

Updated member management

The *Manage members* page has been updated to make it easier to use.

- 48 people are now shown on every page, rather than just 20.
- The *Manage many members* page is now shown when there is more than 48 members in a group, rather than 127.
- The list of people on the *Manage many members* page is now sorted by name.

Sharing links

GroupServer has a *Share* button on each Topic page, next to every post, and on every Image page. It makes it easy to share links on social media sites such as **Twitter**, **Facebook** and **Google+**. It also provides a URL so you can share a link using email, or a different system.

The popup that opens when you click *Share* now it has a *Copy* button, that makes it even easier to share links. Providing the *Copy* button closes [Feature 4176](#).

Web hooks

For a long time GroupServer has used *web hooks* to expose functionality to outside systems. For example, the scripts **mbox2gs**, **smtp2gs**, **senddigest** and the new **sendprofile** (see [Profile status notification](#)) all use web hooks.

Thanks to [Team Z](#) some *generic* web-hooks have been added:

- [Discover](#) all the groups on a site.
- [Add](#) someone to a group.
- [Search](#) for someone by email address.
- [List](#) all the site members.
- [Remove](#) someone from a group.

There is also a new overview of the available hooks (see [Connecting external systems](#)).

Implementing the web-hooks closes [Issue 262](#).

German translation of GroupServer

Far more of the GroupServer user-interface has been translated into German, thanks to the diligent work of Cousin Clara.

Code documentation

The documentation for the low-level system continues to improve in the Limoncello release. Many system now have documentation available on [Read the Docs](#), including all the scripts that are generated during installation. The document components of GroupServer are listed as [sub-projects of GroupServer](#) on Read the Docs.

Minor improvements

- Email notifications should render better in **IBM Notes**, and **Microsoft Outlook** on Windows.
- Some **memory leaks** have been fixed.
- The rewriting of **the Subject** of an email message when the post has been forwarded from another group has been fixed.
- The **WAI-AIRA** roles have been improved, closing [Issue 4156](#).
- An error with a link in the *Unknown email address* notification has been fixed.
- An error with a link to the profile from the *Member has left* notification has been fixed.
- The scripts that use web hooks now handle **301 redirects** correctly, closing [Bug 4162](#).
- Links in email messages can now **use TLS as the protocol** (<https://>) closing [Bug 4171](#). For more information see the document for *Secure connections: TLS, SSL, and HTTPS*.
- Email notifications now use *Hello* as the **opening salutation**, rather than *Dear*.
- A fix for an incorrect **link to Support** in the *Welcome* message that is sent when someone is added to a group has been added.
- A problem with `mailto` links that set a *Subject* failing in Google GMail has been fixed.
- When non-member tries to **unsubscribe** from a group they are now sent an email telling them that they are not a member. Different messages are sent to people with and without profiles.
- Required-widget checking now works if Google Chrome or Opera auto-fill (a.k.a auto-complete) the fields

Get Limoncello

To get Limoncello go to [the Downloads page for GroupServer](#) and follow [the GroupServer Installation documentation](#). Those who already have a functioning installation can [update an existing GroupServer system](#).

Update an Existing GroupServer System

To update a system running the Rakı release of GroupServer (15.03) to Limoncello (15.11) carry out the following steps.

1. Add the `profile_notification_skip` table to the relational database. (The table records those that have chosen to skip the new *Profile status notification*.)
 - (a) Download [the SQL definition of the table](#).

(b) Execute the SQL using the following command:

```
$ psql -U {psql_user} {psql_dbname} -f {filename}
```

Where {psql_user} and {psql_dbname} are the names of the PostgreSQL user and relational-database used by GroupServer (as recorded in `config.cfg`, see [GroupServer installation](#)). The final argument is the name of the SQL file you downloaded (probably `01-skip.sql`).

Alternatively, if you get a password authentication failed error with the above command, the following command may be more successful depending on your database configuration:

```
$ sudo -u postgres psql {psql_dbname} -f {filename}
```

2. Download the Limoncello tar-ball from [the GroupServer download page](#).

3. Uncompress the tar-ball:

```
$ tar xzf groupserver-15.11.tar.gz
```

4. Change to the directory that contains your existing GroupServer installation.

5. Copy the new versions of the configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-15.11/[biv]*cfg .
```

6. Run buildout in your existing GroupServer installation:

```
$ ./bin/buildout -N
```

7. Restart your GroupServer instance (see [Starting and stopping GroupServer](#)).

8. Update **cron** so it sends *the new monthly profile status notification*.

Resources

- Code repository: <https://github.com/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.1.2 GroupServer 15.03 — Rakı with an eye on history

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2015-03-31

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

Introduction

The major *changes to GroupServer* in the Rakı release include the rebuilt email processing sub-system, the ability to export posts, and internationalisation. You can *get Rakı* immediately.

Changes to GroupServer

The most extensive change in Raki is the new *email processing* subsystem. In addition GroupServer can now *export data*, there have been further *improvements to internationalisation*, and the introduction of *message relaying*. Finally, there have been some *minor code improvements*.

Email processing

The components that process email coming into a GroupServer group, and sends email from the group, have been completely changed. This update closes [Feature 387](#).

Receiving email is now handled by code that has been thoroughly unit-tested to ensure that it works with multiple character-sets and encodings. The **checks** to see if a message should be processed have been refactored and are now much more *extensible*, less *coupled*, and more *cohesive*, thanks to [Bill Bushey](#).

The system for **building the email message** before it is sent has been totally rebuilt. Now the message is created using page-templates. This allows the techniques used to customise web pages in GroupServer to be used to customise email messages, including the *prologue* that appears at the top of every message, and at the bottom the *file notifications* and the *footer*.

The **headers** have been refactored to be more extensible. It is now easy to add to the headers, and to change the behaviour of a header. The *Reply-to* header, in particular, has changed to allow the default behaviour of a group to change between

- Reply to the group,
- Reply to the original author, and
- Reply to both.

Finally, the **sending** of an email from a group has been tweaked. However, there *should* be no noticeable change.

Export data

GroupServer has long had the ability to import profile data using the *Add people in bulk* page, and import posts using the `mbox2gs` script. Now with the Raki release GroupServer has the ability to *export profiles* and *export posts*.

Export profiles The new *Export members* page allows an administrator to download a CSV file of all the profile information of the group members. The profile data is from the new `profile.json` view of a profile. The new *Export members* page closes [Feature 4137](#).

Export posts The new *Export posts* page allows an administrator to download an `mbox` file containing the posts (with attachments) made to a group in one particular month. By breaking the archive up like this the administrator is able to download all the posts in the most recent month and add it to an existing `mbox` archive of posts. The new *Export posts* page closes [Feature 3594](#).

Improvements to internationalisation

The effort by [Alice Rose](#) to improve internationalisation support in GroupServer continues apace. GroupServer is now using [Transifex](#) to manage the translations. **You can add translations**, without needing to know any Python or HTML!

All the major components of GroupServer have internationalisation support. This includes the site homepage, the group page, the topic page, and the post page. Adding internationalisation support to the bulk of GroupServer closes [Bug 81](#).

The translation effort is currently focused on French, German, and Spanish (with English being the default language). French is has the most complete localisation, thanks to the amazing efforts of [Razique Mahroua](#).

Finally there is now a [Translation guide](#) included in the [GroupServer](#) documentation. Adding the new internalisation documentation closes [Feature 4031](#).

Message relaying

Messages sent from a GroupServer group can fall afoul of email-servers that enforce [DMARC](#) policies. To overcome this GroupServer rewrites the *From* address: when necessary it replaces the *From* address with one constructed from the unique identifier of the profile of the author. This new address is known as the **obfuscated address** (see [GroupServer 14.06 — Slivovica shots at sunset](#)).

However, the rewritten *From* address has made it difficult to reply the author of the message — and this was particularly problematic when the default *Reply-to* for the group set to the *author* rather than the *group* .

Now, thanks to the work by [Bill Bushey](#) GroupServer will now relay on messages that are sent to an obfuscated address, closing [Feature 4106](#).

Minor code improvements

- The XML for the *Email settings* page has been updated.
- The CSS that is used in the notifications has been fixed.
- The code that makes up the *Topics* list on the group page, the *Topic* and *topic digests* has been refactored, closing [Feature 3739](#).
- The + character (and others) can now be used in email addresses, closing [Bug 3915](#) and [Bug 4036](#).

Get Raki

To get Raki go to the [Downloads](#) page for GroupServer and follow the [GroupServer Installation](#) documentation. Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To update a system running the Calvados release of GroupServer (14.11) to Raki (15.03) carry out the following steps.

1. Download the Raki tar-ball from the [GroupServer download](#) page.
2. Uncompress the tar-ball:

```
$ tar cfz groupserver-15.03.tar.gz
```

3. Change to the directory that contains your existing GroupServer installation.
4. Copy the new version-configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-15.03/[bdiv]*cfg .
```

5. In your **existing** GroupServer installation copy the configuration file to its new location.

(a) Make an `etc` directory:

```
$ mkdir etc/
```

(b) Move the configuration file to the new directory:

```
$ cp parts/instance/etc/gsconfig.ini etc/
```

6. Run buildout in your existing GroupServer installation:

```
$ ./bin/buildout -N
```

7. Restart your GroupServer instance (see [Starting and stopping GroupServer](#)).

Resources

- Code repository: <https://github.com/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.1.3 GroupServer 14.11 — Calvados consumed covertly

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2014-11-25

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by OnlineGroups.net.

Introduction

The major *changes to GroupServer* in the Calvados release include improved internationalisation, and updated email commands. You can *get Calvados* immediately.

Changes to GroupServer

The most extensive change in Calvados is the new *internationalisation*. There are also *updated email commands*, *updated email headers* better *reporting of bouncing email addresses*, and the new *change group-type* page. Finally there have been some *minor code improvements*.

Internationalisation

Alice Rose has lead an effort to add internationalisation to GroupServer. So far internationalisation support has been added to the following products.

- The core page template that lays out the page
- The Join page
- The site homepage

- The *Welcome* area and *Change welcome* page
- The *About* area and *Change about* page
- The *Admin* area
- The *Groups* lists
- The *Topics* list
- The *Change site name* page
- The *Change site timezone* page
- The *Start a group* page
- The *Search for a site member* page
- (Some) of the *Site statistics* page
- The *Change group properties* page

[Status of internationalisation](#) topic in the GroupServer Development group is a good place to follow the improvements to internationalisation. We welcome any and all efforts to improve the internationalisation for GroupServer.

Updated email commands

The system for handling email commands has been completely rewritten and improved. Five commands are currently supported.

Unsubscribe This is the most commonly used command. It removes the group-member that sent the command from the group. See [the `gs.group.member.leave` product](#) for more information.

Subscribe This used to join a **public** group. A person who registers this way will be unable to log in, because he or she will have no password. In addition public groups are rare it has limited use. However, tradition dictates that a `Subscribe` command is supported. See [the `gs.group.member.subscribe` product](#) for more information.

Confirm This message, the companion to `Subscribe`, is used to ensure that a person wants to join a group. See [the `gs.group.member.subscribe` product](#) for more information.

Digest on This is used to switch to the topic-digest. See [the `gs.group.member.email.settings` product](#) for more information.

Digest off This is used to switch to one email per post. See [the `gs.group.member.email.settings` product](#) for more information.

The implementation of the new email commands closes [Bug 3403](#).

Updated email headers

The system for updating email headers has also been completely rewritten. The previous system relied on page-templates defined in the ZMI, which was very hard to update, and harder to test. [The `gs.group.list.sender` product](#) now defines the fourteen email headers that are deleted, added, or modified. The implementation of the new headers closes [Bug 267](#).

In addition the various `List` headers have been added, which closes [Issue 3785](#).

Reporting of bouncing email addresses

When an email message cannot be delivered often a *bounce* message will be set back to the system that sent the message. For many years GroupServer has kept track of these bounce messages; if an address of a group-member causes bounces on five separate days in a 60 day window then that email address is set to *unverified*.

The new *Bounces* page now shows the group administrator a record of the bouncing email addresses. In addition a notification is sent to the administrator when the email address of a group member is disabled. If the member has any additional email addresses then notifications is sent to these addresses also.

The updating of the bounce-handling system closes [Feature 3614](#), [Feature 3772](#) and [Feature 3773](#).

Change group-type

There are four types of group supported by GroupServer:

- Discussion groups such as [GroupServer development](#),
- Announcement group such as [GroupServer announcements](#),
- Support groups such as support@onlinegroups.net, and
- Closed groups.

Calvados introduces the *Change group-type* page, which allows the type of group to be quickly and easily changed, closing [Feature 702](#).

Minor code improvements

- The [primary code repository](#) for GroupServer is now at GitHub. We hope that this will make GroupServer easier for others to work on.
- The *Change image* page is now in its own product, `gs.profile.image.edit`, closing [Feature 601](#)
- The notifications that are sent out when someone leaves a group have been updated, closing [Feature 4061](#).
- The subject-line prefix is correctly set when changing the general group properties, closing [Bug 640](#).
- Many products now have [Sphinx](#) documentation. This includes [the documentation for GroupServer installation](#).

Get Calvados

To get Calvados go to [the Downloads page](#) for GroupServer and follow [the GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To update a system running the Slivovica release of GroupServer (14.06) to Calvados (14.11) carry out the following steps.

1. Download the Calvados tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball:

```
$ tar cfz groupserver-14.11.tar.gz
```

3. Change to the directory that contains your existing GroupServer installation.

4. Copy the new version-configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-14.11/[biv]*cfg .
```

5. In your existing GroupServer installation run:

```
$ ./bin/buildout -n
```

6. Restart your GroupServer instance.

TODO Update the table with the confirmation IDs for the new subscription command

Resources

- Code repository: <https://github.com/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.1.4 GroupServer 14.06 — Slivovica shots at sunset

Authors Michael JasonSmith;

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2014-06-03

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

Introduction

The major *changes to GroupServer* in the Slivovica release include making the system easier to install, better at handling email, and improving the reliability. You can *get Slivovica* immediately.

Changes to GroupServer

For new installations of GroupServer the most important change is the *updated buildout recipes*, that should make the process of installing GroupServer more robust. For existing systems *DMARC support* has been added, as well as improvements to how *incoming SMTP* is handled. Finally there have been many other *minor code improvements*.

Updated buildout recipes

GroupServer itself is installed using four *recipes* that are used by Buildout. All four recipes have been improved. There is now a base recipe *class* that is shared by the recipes, and unit-tests have been added to ensure the recipes work as expected. Specific improvements include the following.

- The recipe that creates the Postfix configuration has been updated, fixing a [problem that Bill found](#).
- The recipe that creates the PostgreSQL tables now uses the [setuptools](#) package to extract the SQL files from the different modules. The modules are now passed into the recipe from Buildout, rather than being written in the recipe.

- The creation of the GroupServer site is now simpler, as only the site-administrator is created during installation. This solves many issues with the email-address of the normal group-member conflicting with the email address of the site administrator.

DMARC Support

With the Slivovica release of GroupServer messages are more likely to be received by people whose email is provided by hosts that implement [strict DMARC](#). With strict DMARC messages are checked to ensure that message was actually sent from the host listed in the `From` address. All mailing-lists (including GNU Mailman, Google Groups, and Yahoo! Groups) modify the message before sending it out, and this would cause the DMARC check to fail.

Like most other mailing lists, GroupServer now rewrites the `From` address if the group member uses an email provider that implements strict DMARC. The new address includes the name of the sender, and the ID of his or her profile (see [Feature 4108](#)). This ensures that the person who sent the message can still be identified, and email message will still get through.

Incoming SMTP

With the Slivovica release of GroupServer email messages are correctly sent to GroupServer when it is running on a non-standard port. Messages are received by Postfix, which then uses the `smtp2gs` programme to add the message to GroupServer. Previously manual intervention was needed if GroupServer was listening to a non-standard port.

Now all components of GroupServer correctly handle the port that is set in the configuration file when GroupServer is first set-up. This includes the *recipe* that creates the alias file for Postfix, the `smtp2gs` script that uploads the email message, and the `gs.form` library that `smtp2gs` uses to connect to GroupServer.

Minor code improvements

Four components have undergone minor updates for Slivovica.

Add member: Fixed an error with the dependencies of the product that added a member to a group.

Site style: Some errors in the CSS have been fixed.

mbox importing: The code that imports `mbox` files has been updated, and it now uses the same core code as the `smtp2gs` script.

Starting public groups: Public groups are now visible on the homepage of the site as soon as they are started, closing [Bug 4105](#).

In addition the following general changes have been made to GroupServer.

- Three fundamental *products* — `gs.core`, `gs.dmarc`, and `gs.form` — have been **published** to PyPI.
- [Sphinx](#) is now used to generate the **documentation** for `gs.config`, `gs.core`, `gs.database`, `gs.dmarc`, `gs.form`, and `gs.group.messages.add.smtp2gs`.
- **Unit tests** have been added to many of the products that make up GroupServer. These tests make it more likely that these components that make up GroupServer will work correctly now and in the future.
- Where possible **Python 3 support** has been added to the products that make up GroupServer. The main change has been to switch to Unicode-text by default. While not all products have been updated, many have.
- **Strict mode** has been turned on in more of the JavaScript modules that support the GroupServer interface. Code written with strict-mode turned on is more likely to be correct, and is executed more quickly in modern browsers.
- Finally, more code now raises errors, rather than relying on `assert` statements. As these errors should never happen hopefully no-one will notice a difference.

Get Slivovica

To get Slivovica go to [the Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To update a system running the Ouzo release of GroupServer (14.03) to Slivovica (14.06) carry out the following steps.

1. Download the Slivovica tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball:

```
$ tar cfz groupserver-14.06.tar.gz
```

3. Change to the directory that contains your existing GroupServer installation.
4. Copy the new version-configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-14.06/[vz]*cfg .
```

5. In your existing GroupServer installation run:

```
$ ./bin/buildout -n
```

6. Restart your GroupServer instance.

Resources

- Code repository: <https://source.iopen.net/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.1.5 GroupServer 14.03 — Ouzo utilised as an unguent

Authors Michael JasonSmith; Bill Bushey; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2014-03-20

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 4.0 International License](#) by [OnlineGroups.net](#).

Introduction

It has been seventeen months since the last release of GroupServer (Absinthe, 12.11). The major delay was caused by the creation of a new user-interface for GroupServer. Almost all the *changes to GroupServer* are user-visible — and include a new notifications, and updated systems for adding people. You can *get Ouzo* immediately.

Changes to GroupServer

The most significant change to GroupServer in the Ouzo release is the *new style pages*. In addition the system for *adding people* in bulk has been completely rewritten, there has been an extensive *notifications update*, and *other updates*.

New style pages

The main update to GroupServer has been the implementation of new pages on the Web interface ([Issue 3423](#)). The *navigation* has been extensively reworked: there is an *updated layout* for most pages, and pages have been integrated with *Twitter Bootstrap*². The *profile images* are now (mostly) square, with new code to improve their performance. In addition the pages have been made more *responsive*, and the *performance* of the pages has been improved.

Acknowledgements The pages were redesigned by Mike Harding from [Cactuslab](#), to whom all praise should be given.

Navigation The navigation scheme for GroupServer has changed dramatically in this release, moving away from the context-menu (along the left), and dropping the site-menu (along the top) entirely. In their place is a system based on breadcrumbs ([Issue 3510](#)).

The *Profile* page is the only page left that makes use of the context-menu.

Updated layout The *Group* page³ and *Site* homepage⁴ have been updated so they have asymmetric columns — with less important information placed in the smaller column on the right. Both make use of *Twitter Bootstrap*, the updated *profile images*, and the *updated lists*.

The *Group* page gets an updated *About* area, and *Change about* page ([Issue 3871](#)). The *Site* homepage gets a new *Welcome* area ([Issue 3869](#)) in addition to an updated *Introduction* ([Issue 3870](#)).

The layout of the *Topic* page and *Post* page have been updated to move the metadata (such as who posted the message, and when) to the left of the post ([Issue 3455](#)).

The *Image* page, which displays an image that has been posted to a group, has been updated to reflect the layout of the *Topic* page. Problems with SVG images have also been resolved ([Issue 3663](#)).

Most of the other pages have been updated to take *Twitter Bootstrap* and the new *navigation* into account.

Twitter Bootstrap The GroupServer user-interface is now based on *Twitter Bootstrap 2.3*. The look of every single page has changed significantly as a consequence.

The tabs, on the *Group* page and *Site* page, have dropped the use of jQuery.UI tabs in favour of the Bootstrap Tabs. The Popover class is used to show information including the privacy of a group and posts, encouragement for the new administrator, and the Share widget.

Profile images Another major change has been with the profile images. They are now shown in more places and at *multiple resolutions*. In addition most of the images are *square*. If a group member has not set an image a *missing profile image* image is shown.

² The product for supplying the CSS is located at <<https://source.iopen.net/groupserver/gs.content.js.bootstrap>>

³ The code responsible for laying out the *Group* page is provided by <<https://source.iopen.net/groupserver/gs.group.home>>

⁴ The code responsible for laying out the *Site* homepage is provided by <<https://source.iopen.net/groupserver/gs.site.home>>

Multiple resolutions The profile-images in releases of GroupServer prior to Ouzo were restricted to 81×108 pixels. Now the content provider ⁵ is able to supply profile images at a range of resolutions ([Issue 588](#)). In addition the *image compression* enhancements have also been rolled out to the profile image, so small images are now directly embedded into the page.

Square The profile images are now square by default ⁶. These square images are shown in more places, such as in the new *Recently active* list on the *Group* page ([Issue 3873](#)), the *Profile* link at the top of every page, and the *Post message* area of the *Topic* and *Start topic* pages.

Updated lists At its core GroupServer provides lists of things: groups, topics, posts, files, and people. All these lists have been updated to improve the hierarchy of information.

- The *Groups* list on the *Site* page now lists all the visible groups, making the *Groups* page redundant ([Issue 3449](#)).
- The *Files* icons for a topic are no-longer shown by default. Instead a single attachment-icon is shown, and icons for the files are displayed in a tooltip.
- Who made the most recent post, and when, is more prominently displayed.
- The topic-keywords are labelled *Keywords*.
- There is a new list of *Recently active* members on the *Group* page, which is loaded through AJAX ([Issue 3873](#)).
- The list of *Files* has been moved to the secondary column of the *Group* page, from the main area.

Responsive The GroupServer web pages are now more responsive to the size of device ([Issue 3909](#)). This allows the pages to look good from screens found on desktops, down to small feature phones. This allows people to keep up with conversations anywhere and any time. Some pages, such as the *Image* page ([Issue 3508](#)) have had particular attention to ensure the page works well at multiple sizes.

Performance The performance Web interface has been massively improved. The primary way of doing this has been with *refactored JavaScript*. In addition *image compression* has been increased and *font icons* introduced. Finally, many small changes have been made to the layout of the pages to reduce the *time to glass*.

Refactored JavaScript All the JavaScript used by GroupServer has been refactored into separate modules ([Issue 344](#)). This makes documentation and maintenance far easier, at the expense of speed. To compensate, all JavaScript (including that supplied by jQuery ⁷ and Twitter Bootstrap ⁸) is deferred until after the page has been shown ⁹. In addition, all the JavaScript has been *minified* to reduce the amount of data that is transported, and to speed the parsing by the Web browser. Finally, almost all the JavaScript is *asynchronously* loaded ¹⁰.

The JavaScript code loads and assists with navigating the lists of recent topics, posts and files. This code has been refactored so the all the lists share the same code ([Issue 3507](#)). In addition the lists are only loaded when the corresponding tab is visible. Combined this greatly reduces the number of requests required to load the page.

Strict mode has been enabled for all the core JavaScript modules, and some of the other modules. This has the combined effect of reducing the number of errors, and improving performance by allowing the browser to optimise the code.

⁵ The product for supplying the profile-image content provider is located at <<https://source.iopen.net/groupserver/gs.profile.image.base>>

⁶ The product for supplying the square profile-image is located at <<https://source.iopen.net/groupserver/gs.profile.image.square>>

⁷ The product for supplying the jQuery JavaScript is located at <<https://source.iopen.net/groupserver/gs.content.js.jquery.base>>

⁸ The product for supplying the Twitter Bootstrap code is located at <<https://source.iopen.net/groupserver/gs.content.js.bootstrap>>

⁹ The product for determining what JavaScript is loaded, and how, is provided by <<https://source.iopen.net/groupserver/gs.content.layout>>

¹⁰ The code for asynchronously loading the JavaScript is provided by <<https://source.iopen.net/groupserver/gs.content.js.loader/>>

Image compression Many JPEG images that are posted by group members have very low compression (or very high quality, depending how you like to look at it). GroupServer now produces thumbnails with aggressive compression by default ([Issue 663](#)). As the dimensions of the image are reduced the quality of the image is also dropped — making the images far smaller ¹¹. When images are *particularly* small a data-URI is used to directly embed the image into the page, reducing the need to make an HTTP request to fetch the image.

Font icons The new user-interface uses a font to provide the different icons in the interface ([Issue 3788](#)). These are quick to load and render. In addition they are independent of the resolution of the device, so work well in the new *responsive* user interface.

Adding people

The system for adding people to a group has undergone many improvements, primarily to the pages that allow the new member details to be uploaded by CSV, but also to the system that allows existing site members to be invited to join a group.

The systems for inviting people in bulk ¹² — and adding people in bulk ¹³ — using a CSV file have been completely rewritten ([Issue 3494](#)). Both systems now use AJAX and JSON to parse the CSV file, and invite ¹⁴ or add ¹⁵ individual people. This, combined with widgets provided by *Twitter Bootstrap*, allows for continual progress updates, avoids server timeouts, and is **massively** more usable.

On a more minor note, the page for inviting site members to join a group has been enhanced with the addition of email-addresses, which helps distinguish between people with similar names ([Issue 452](#)).

Notifications update

Most of the notifications in GroupServer have been updated in Ouzo ([Issue 3892](#)). *Premailer* is now used to embed CSS into the HTML-formatted messages. The CSS itself is specified using a system of “skins” that is very similar to what is used for the Web pages ¹⁶, and the default look is very similar to the Web user interface ¹⁷.

Almost all the notifications have been moved to the file-system — rather than requiring templates in the ZMI ([Issue 269](#)) — and the use of the new notifications is now far more consistent than before.

Topic digests The system for producing the topic digests ¹⁸ has been completely rewritten ([Issue 408](#)). The digest email is provided in both HTML and plain-text formats. In addition the code for sending the digests ¹⁹ has been rewritten, so it uses less memory, and is more secure ([Issue 3415](#) and [Issue 3417](#)).

Other updates

As well as the major rewrite of the user-interface a number of smaller changes have been made with the Ouzo release that makes GroupServer more useful, more usable, and easier to maintain.

¹¹ The code for determining how images should be displayed is provided by `<https://source.iopen.net/groupserver/gs.image>`

¹² The code for inviting people by uploading a CSV file is provided by `<https://source.iopen.net/groupserver/gs.group.member.invite.csv>`

¹³ The code for adding people by uploading a CSV file is provided by `<https://source.iopen.net/groupserver/gs.group.member.add.csv>`

¹⁴ The code for inviting someone by JSON is provided by `<https://source.iopen.net/groupserver/gs.group.member.invite.json>`

¹⁵ The code for adding someone by JSON is provided by `<https://source.iopen.net/groupserver/gs.group.member.add.json>`

¹⁶ The code for specifying the CSS for the HTML-formatted notifications is provided by `<https://source.iopen.net/groupserver/gs.content.email.css>`

¹⁷ The code for specifying the layout of the messages is provided by `<https://source.iopen.net/groupserver/gs.content.email.layout>`

¹⁸ The system for generating the topic digests is provided by `<https://source.iopen.net/groupserver/gs.group.messages.topicsdigest>`

¹⁹ The system for sending the topic digests is provided by `<https://source.iopen.net/groupserver/gs.group.messages.senddigest>`

Autocomplete with *Start a topic*: The *Start a topic* page²⁰ now has type-ahead (provided by *Twitter Bootstrap*) that suggests the names of existing topics in the group (*Issue 282*).

Show password: All password entries (for login and setting a password) now have a toggle to allow a group member to hide his or her password in public places (*Issue 519*)²¹.

Attachment detection: The code for determining the attachments to show, and the attachments to hide, has been rewritten to allow more attachments through (*Issue 4073*).

Join and leave log: The *Join and leave log* has been moved to the *Members* page (*Issue 3683*).

WYEditor updated: The *WYEditor* is used to provide editing of HTML content such as the *About* area in a group. It has been updated to work with jQuery 1.9 (*Issue 3868*).

Keywords on the *Topic* page: Keywords, summarising what has been discussed, are now shown at the top of the *Topic* page (*Issue 877*).

Privacy on the *Group* page: The privacy setting for a group is shown on the *Group* page, near the email address for the group (*Issue 3914*).

Accessibility: WAI-ARIA attributes have been added throughout GroupServer to improve the accessibility.

Python 3 updates: The slow journey to convert GroupServer from Python 2 to Python 3 has been started. At this stage three sets of changes have been made, or are being made:

- Ensuring the code is consistent with [PEP-8](#),
- Switching to Unicode literals where possible ([PEP-3112](#)), and
- Switching to absolute import ([PEP-328](#)).

Get Ouzo

To get Ouzo go to the [Downloads](#) page for GroupServer and follow *the GroupServer Installation documentation*. Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To update a system running the Absinthe release of GroupServer (12.11) to Ouzo (14.03) carry out the following steps.

1. Download the Ouzo tar-ball from the [GroupServer download page](#).
2. Uncompress the tar-ball:

```
$ tar cfz groupserver-14.03.tar.gz
```

3. Change to the directory that contains your existing GroupServer installation.
4. Make a backup of your custom configuration:

```
$ cp custom.cfg custom-bk.cfg
$ cp config.cfg config-bk.cfg
```

5. Copy the new configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-14.03/*.cfg .
```

6. Restore your custom configuration:

²⁰ The *Start a topic* page is provided by <https://source.iopen.net/groupserver/gs.group.messages.starttopic>

²¹ The toggle to show or hide a password is provided by <https://source.iopen.net/groupserver/gs.profile.password>

```
$ mv custom-bk.cfg custom.cfg
$ mv config-bk.cfg config.cfg
```

7. In your existing GroupServer installation run:

```
$ ./bin/buildout -n
```

8. Restart your GroupServer instance.

Resources

- Code repository: <https://source.iopen.net/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.1.6 GroupServer 12.11 — Absinthe acquired arbitrarily

Authors Michael JasonSmith; Richard Waid; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2012-11-12

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by OnlineGroups.Net.

Contents

- *GroupServer 12.11 — Absinthe acquired arbitrarily*
 - *Introduction*
 - * *Acknowledgements*
 - *Changes Visible to Participants*
 - * *New Search System*
 - * *Better Error Pages*
 - * *New Help System*
 - * *In-Context Administration Guide*
 - * *Closed Groups*
 - * *Profile Search*
 - * *SVG Thumbnails*
 - *Changes to the Underlying System*
 - * *Configuration System*
 - * *Improved Email Handling*
 - * *New Authentication System*
 - * *Relstorage*
 - * *Performance Improvements*
 - * *SQLAlchemy Update*
 - * *More Flexible Group Page*
 - *Get Absinthe*
 - * *Update an Existing GroupServer System*
 - *Resources*

Introduction

There are fourteen major changes to GroupServer in the Absinthe release — making the system more useful, usable and extensible for both group members, administrators, and developers.

The *changes visible to participants* are mostly subtle improvements to existing features, rather than new features. In contrast, the *changes to the underlying system* are complex and extensive.

You can *get Absinthe* immediately.

Acknowledgements

Thanks to [Bill Bushey](#), with the support of [E-Democracy.org](#), for the improvements to the *SVG thumbnails*. Thanks to Bill and [Marek Kuziel](#) for testing some early versions of Absinthe.

Changes Visible to Participants

The most visible change for the participants is a *new search system*. Some *better error pages* have also been created. To help participants and administrators there is a *new help system*, while an *in-context administration guide* will help administrators get their new group going more easily. Administrators now have the ability to create *closed groups*, and the new *profile search* will help administrators find the profiles of participants. Finally, *SVG thumbnails* are now shown.

New Search System

The new search system is the most visible change in Absinthe.

- The *Topics* tab on both the site homepage and the group page have a *Search* entry ²².
- The *Posts* tab in a group has as *Search* entry also, allowing posts to be searched ²³.
- Clicking a topic keyword will search for other topics with that keyword ²⁴.
- Searching now uses *stems*; searching for `search` will match topics that contain the words `searches`, `searching` or `searched`, for example.
- Searching is faster, as the full-text retrieval features of [PostgreSQL](#) are used ²⁵.
- The display of topics in a site or group is much faster, as the keywords are now generated when someone posts, rather than when the topic is loaded.
- Searching using non-ASCII characters now works ²⁶.

Better Error Pages

The *Permission Denied* page has been improved to add some suggestions about what the participant should do ²⁷. In addition the `mailto` that is embedded in the page now includes the URL of the problem page in the body of the email message to the Support address.

²² The *Search* box that used to appear on every page has now been removed, as it is easier to use the *Search* entry in the *Topics* tab. This closes Bug 3434.

²³ Searching in posts closes [Feature 3497](#).

²⁴ Creating topic keywords that can be clicked closes [Feature 878](#).

²⁵ Using the full-text retrieval feature of [PostgreSQL](#) closes [Feature 224](#).

²⁶ Being able to search for non-ASCII characters closes [Bug 603](#).

²⁷ Updating the *Permission Denied* page closes [Bug 646](#).

The Unexpected Error (500) page and the Not Found (404) page now work with `infrae.wsgi`, for systems that run GroupServer behind a WSGI front-end. Neither of these error-pages redirect to display the error.

New Help System

The old monolithic manuals have been replaced. The new more dynamic system will automatically show help for the features that are installed in GroupServer, including the custom features. Some of the old pages have been retained and updated.

In-Context Administration Guide

To help get groups started there is a new system to encourage the group administrator²⁸. The current advice includes:

- Start a topic
- Invite people,
- Write some text in the *About* tab, and
- Make a group Private (rather than Secret).

Closed Groups

Administrators can now close groups²⁹. There are two reasons for needing to do this.

1. The group may be *starting*, and the administrator does not want the members to post until everyone is in the group.
2. The group may be *finished*, and the administrator does not want any more posts to be made, but he or she still wants the archive available.

While there is no front-end user interface that allows the group-type to be changed³⁰, an administrator can change the type of the group by making a change in the ZMI.

Profile Search

A simple profile search has been added³¹. A more complex search system has not been added because the privacy issues are still being resolved.

SVG Thumbnails

GroupServer now correctly displays a thumbnail of an SVG image at the bottom of each post³². This item was picked from the list of [low hanging fruit](#), where there are other (relatively) strait forward tasks listed.

²⁸ Creating the encouragement closes [Feature 3501](#) and [Feature 177](#).

²⁹ Creating the closed-group closes [Feature 449](#).

³⁰ The issue for creating a selectable group type is [Feature 702](#).

³¹ The creation of a basic profile search closes [Feature 3486](#).

³² Handling SVG Thumbnails closes [Bug 635](#).

Changes to the Underlying System

We have made significant changes to the underlying GroupServer system in the Absinthe release. The system will be easier to maintain because of a new *configuration system*, *improved email handling*, and a *new authentication system*. Installation is simpler because *relstorage* is used by default. There have also been some significant *performance improvements*. Developers will notice the *SQLAlchemy update*, and a *more flexible group page*.

The changes to the underlying system have been so extensive that we have decided to change the naming scheme for the GroupServer releases. The new releases belong to the **Awesome Aperitifs** series. Hence this *Absinthe Acquired Arbitrarily* release. Internally, the eggs in this series are given the version 2.0 (which you can see by looking at the `versions.cfg` file in the build directory). The version-number of the *release* will continue to use the `month.day` format. The old series was known as **Frozen Treats**; *Falodeh Consumed with an Eye on History* was the aptly-named last release in that series. (Eggs in the Frozen Treats series were given the 1.0 version.)

Configuration System

Administration is now simpler, especially for production systems, as the configuration for important parts of GroupServer are now in a file that is external to the ZODB. The new configuration system handles the database, the *improved email handling*, and the *new authentication system*. It is based on a INI file, located in `parts/instance/gsconfig.ini`.

Improved Email Handling

The email-handling subsystem of GroupServer has been completely rewritten. Changes have been made to both the handling of outgoing mail and incoming mail.

The setup for the **outgoing** SMTP system has moved from the ZODB (accessed through the ZMI) to an INI file ³³, thanks to the new *configuration system*. Documentation for the new outgoing SMTP system can be found in the [README for the gs.email product](#).

The script that is used to add email messages to a group, the **incoming** SMTP system, has been rewritten ³⁴. It is now easier to use, better documented, and works. Documentation for the new script can be found by running `./bin/smtp2gs -h` or reading the [README for the gs.group.messages.add.smtp2gs product](#).

New Authentication System

A new authentication system has been created, for the server-side scripts ³⁵. These scripts, such as the those involved in the *improved email handling*, now pass a *token* to the server when they carry out tasks. This eliminates the need to store the password of the administrator in various plain-text files.

Relstorage

By default the [Relstorage product](#) is now used to store the ZODB. This system stores the pickled objects in a relational database, rather than in the file-system. (The [PostgreSQL](#) database is used by GroupServer.) This allows greater scalability, without the need to separately install Zope Enterprise Objects (ZEO).

³³ Moving the SMTP configuration to an INI file means that the two `MailHost` instances from the ZODB can be removed, which closes [Bug 365](#).

³⁴ Rewriting the script that is used to add email messages to a group closes [Feature 687](#) and [Feature 3536](#).

³⁵ The creation of a new authentication system closes [Bug 3416](#).

Performance Improvements

There have been some major performance improvements made to GroupServer in the Absinthe release. This includes the removal of some old poorly performing code ³⁶, and altering some of the member management code ³⁷.

SQLAlchemy Update

The entire interface between GroupServer and the PostgreSQL relational database has been rewritten ³⁸. This has allowed GroupServer to update its SQLAlchemy dependency from the ancient 0.3 release to the current 0.7 release.

More Flexible Group Page

The group page was refactored to make it more flexible ³⁹. This allows the addition of the *in-context administration guide*, and for other features to be added by skins.

Get Absinthe

To get Absinthe go to the [Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

Updating a system running the Faloodah release of GroupServer (12.06) to Absinthe is a three-step process, which includes updating the relational database, the products, and the scripts.

Relational Database

The biggest change that is needed to update GroupServer to the Absinthe release is to update the relational database, to support the *new search system* and some of the *performance improvements*. The tables that are used to store the posts, topics and topic keywords all need to be updated.

First, create a backup. While every effort has been made to create a upgrade path that is smooth and with low risk, there is still a chance that something can go wrong. As such it is prudent to **create a backup**. First, **create a backup** of the relational database:

```
$ pg_dump -U gsadmin groupserver > gs-backup.sql
```

Where `gsadmin` is the PostgreSQL user that you set up when installing GroupServer, and `groupserver` is the name of the database.

If you use *relstorage*, **create a backup** of the ZODB:

```
$ pg_dump -U gszodbadmin groupserverzodb > gs-zodb-backup.sql
```

Where `gszodbadmin` is the PostgreSQL user for *relstorage* that you set up when installing GroupServer, and `groupserverzodb` is the name of the database.

Posts

Note Update the `posts` table **after** you create a backup.

³⁶ The removal of the old `division_object` getter closes [Bug 279](#).

³⁷ The optimisation of the member-handling code closes [Bug 3659](#).

³⁸ An unintentional site-effect of rewriting the database interface was a fix for [Bug 203](#).

³⁹ As part of the update to the group, [Feature 419](#) was closed.

Begin by updating the table that stores the posts.

1. Log in to the [PostgreSQL](#) command line:

```
$ psql -hlocalhost -Ugsadmin groupserver
```

Where `gsadmin` is the PostgreSQL user that you set up when installing GroupServer, and `groupserver` is the name of the database.

2. Alter the `post` table to add the full-text retrieval (FTR, or full-text search, FTS) column, by executing the following SQL:

```
ALTER TABLE post ADD COLUMN fts_vectors tsvector;
```

3. Update the rows of the `post` table to add the FTR data. This may take some time:

```
UPDATE post
SET fts_vectors = to_tsvector('english',
                             left(coalesce(subject, '') || ' ' || coalesce(body, ''),
                                   1048575));
```

4. Create an index for the FTR data. This may take some time:

```
CREATE INDEX post_fts_vectors ON post USING gin(fts_vectors);
```

5. Create an index for the posts, sorted by the last post date:

```
CREATE INDEX post_last_post_date_idx ON post (date DESC);
```

6. Create a trigger to update the FTR data whenever a new post is made:

```
CREATE TRIGGER fts_vectors_update
BEFORE INSERT or UPDATE ON post
FOR EACH ROW EXECUTE PROCEDURE
    tsvector_update_trigger(fts_vectors, 'pg_catalog.english', subject,
                           body);
```

Topics

Because people search topics as well as posts the FTR information needs to be present in both tables.

1. Add the FTR column to the `topic` table:

```
ALTER TABLE topic ADD COLUMN fts_vectors tsvector;
```

2. Drop the old trigger:

```
DROP TRIGGER count_word_count_rows ON word_count;
```

3. Drop the old tables that were used for searching:

```
DROP TABLE topic_word_count;
DROP TABLE word_count;
```

Even with the FTR data duplicated in the `post` and `topic` table, there is a nett saving of space once these two tables are dropped.

4. Add the function that is used to create the *body* of the topic:

```
CREATE OR REPLACE FUNCTION topic_body (topic_id TEXT)
RETURNS TEXT AS $$
DECLARE
    topic_text TEXT;
```

```

        subject TEXT;
        retval TEXT;
BEGIN
    SELECT string_agg(post.body, ' ') INTO topic_text
        FROM post
        WHERE post.topic_id = topic_body.topic_id
            AND post.hidden IS NULL;
    SELECT COALESCE(post.subject, '') INTO subject
        FROM post WHERE post.topic_id = topic_body.topic_id LIMIT 1;
    retval := left(subject || ' ' || topic_text, 1048575);
    RETURN retval;
END;
$$ LANGUAGE 'plpgsql';

```

5. Create the function that will topic table with FTR data:

```

CREATE OR REPLACE FUNCTION topic_ftr_populate ()
RETURNS void AS $$
DECLARE
    total_topics REAL;
    trecord RECORD;
    topic_vector tsvector;
    topic_text TEXT;
    i REAL DEFAULT 0;
    p REAL;
BEGIN
    SELECT CAST(total_rows AS REAL) INTO total_topics
        FROM rowcount WHERE table_name = 'topic';
    FOR trecord IN SELECT * FROM topic WHERE fts_vectors IS NULL LOOP
        RAISE NOTICE 'Topic %', trecord.topic_id;
        topic_vector := to_tsvector('english', topic_body(trecord.topic_id));
        UPDATE topic SET fts_vectors = topic_vector
            WHERE topic.topic_id = trecord.topic_id;
        i := i + 1;
        p := (i / total_topics) * 100;
        RAISE NOTICE ' Progress % %%', p;
    END LOOP;
END;
$$ LANGUAGE 'plpgsql';

```

6. Populate the topic table with FTR data. This may take some time:

```
SELECT topic_ftr_populate();
```

7. Create an index for the FTR data. This may take some time:

```
CREATE INDEX topic_fts_vectors ON topic USING gin(fts_vectors);
```

8. Create an index for the topics, sorted by the last post date:

```
CREATE INDEX topic_last_post_date_idx ON topic (last_post_date DESC);
```

9. Create a trigger to update the FTR data:

```

CREATE OR REPLACE FUNCTION topic_fts_update ()
RETURNS TRIGGER AS $$
DECLARE
    topic_text TEXT;
BEGIN
    topic_text := topic_body(NEW.topic_id);

```

```
        NEW.fts_vectors := to_tsvector('english', topic_text);
        RETURN NEW;
    END;
$$ LANGUAGE 'plpgsql';
CREATE TRIGGER topic_update_trigger_01
    BEFORE INSERT OR UPDATE ON topic
    FOR EACH ROW EXECUTE PROCEDURE topic_fts_update ();
```

Topic Keywords

Finally, the system that displays the topic keywords has been changed. The keywords are now calculated when someone posts, and are stored in the `topic_keywords` table. Previously they were calculated when the list of topics was displayed.

1. Download the file `03-keywords.sql`:

```
wget --no-check-certificate https://source.iopen.net/groupserver/gs.group.messages.topic/rawfile
```

This contains the SQL that is normally executed when Absinthe is installed.

2. Interpret (execute) the file in PostgreSQL:

```
\i /path/to/the/download/03-keywords.sql
```

Where `/path/to/the/download` is the full path to where the file `03-keywords.sql` is stored.

3. Create the function to populate the new `topic_keywords` table:

```
CREATE OR REPLACE FUNCTION topic_keywords_populate ()
    RETURNS void AS $$
    DECLARE
        total_topics REAL;
        trecord RECORD;
        topic_text TEXT;
        new_keywords TEXT[];
        i REAL DEFAULT 0;
        p REAL;
    BEGIN
        SELECT CAST(total_rows AS REAL) INTO total_topics
            FROM rowcount WHERE table_name = 'topic';
        FOR trecord IN SELECT * FROM topic LOOP
            RAISE NOTICE 'Topic %', trecord.topic_id;
            topic_text = topic_body(trecord.topic_id);
            SELECT ARRAY(SELECT word
                           FROM topic_keywords(trecord.topic_id, topic_text))
                INTO new_keywords;
            INSERT INTO topic_keywords VALUES(trecord.topic_id, new_keywords);
            i := i + 1;
            p := (i / total_topics) * 100;
            RAISE NOTICE ' Progress % %%', p;
        END LOOP;
    END;
$$ LANGUAGE 'plpgsql';
```

4. Populate the `topic_keywords` table. This may take some time:

```
SELECT topic_keywords_populate();
```

Products

To update an existing GroupServer installation to Absinthe carry out the following steps.

1. Download the Absinthe tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball:

```
$ tar cfz groupserver-12.11.tar.gz
```

3. Change to the directory that contains your existing GroupServer installation.
4. Make a backup of your custom configuration:

```
$ cp custom.cfg custom-bk.cfg
$ cp config.cfg config-bk.cfg
```

5. Copy the new configuration files to your existing GroupServer installation:

```
$ cp ../groupserver-12.11/*.cfg .
```

6. Restore your custom configuration:

```
$ mv custom-bk.cfg custom.cfg
$ mv config-bk.cfg config.cfg
```

7. Disable the creation of the database tables:

```
$ echo 1 > var/create-tables.cfg
```

8. Disable the creation of a new GroupServer site:

```
$ echo 1 > var/setup-gs.cfg
```

9. In your existing GroupServer installation run:

```
$ ./bin/buildout -n
```

10. Restart your GroupServer instance.

Scripts

Some external scripts have changed in the Absinthe release of GroupServer, and need to be changed. In addition some ZMI scripts should also be updated.

The script `smtp2zope` used to be used to marshal an email message from Postfix into GroupServer. With the *improved email handling* this script should be deleted. The replacement script is called `smtp2gs`. It will be created when you update the products. The command is simpler to use than the old script; the options for the script are shown by running:

```
$ ./bin/smtp2gs --help
```

Alternatively, [the README for the `gs.group.messages.add.smtp2gs` product](#) documents the options.

The directory `potfix_config` in your GroupServer installation will contain an example aliases file for Postfix that uses `smtp2gs`. This can be used to replace the old calls to GroupServer from Postfix.

ZMI Scripts

Two scripts in the ZMI have to be replaced to gain some of the significant *performance improvements*.

1. Visit the ZMI for your site. By default it is at `<http://localhost:8080/manage>`.
2. Go to the folder `/example/ListManager`.
3. Select the `xwf_email_header` script.
4. Replace the contents of the script with the following:

```
groupId = list_object.listId()
siteId = list_object.siteId
site = getattr(context.Content, siteId)
group_object = getattr(site.groups, groupId)

# we copy the property sheet, because we won't be able to access it
# in the lower layer
group_properties = {}
for p in group_object.propertyItems():
    group_properties[p[0]] = p[1]
group_properties['id'] = group_object.getId()

xmailer = getValueFor('xmailer')
mailto = getValueFor('mailto')
replyToProp = list_object.getProperty('replyto', '')
if replyToProp == 'sender':
    replyto = None
else:
    replyto = getValueFor('mailto')

try:
    group_properties['division_id'] = site.aq_explicit.getId()
    group_properties['division_title'] = site.aq_explicit.title
except:
    group_properties['division_id'] = ''
    group_properties['division_title'] = ''

files = []
try:
    storage = context.FileLibrary2.get_fileStorage()
    for file_id in file_ids:
        file = storage.get_file(file_id)
        if file:
            files.append(file)
except:
    pass

return context.email_header(REQUEST, list_object=list_object,
                             group_properties=group_properties,
                             getValueFor=getValueFor, title=title, mail=mail,
                             body=body, files=files, post_id=post_id,
                             mailto=mailto, replyto=replyto,
                             xmailer=xmailer).strip()
```

5. Click the Save Changes button.
6. Click ListManager (at the top of the page) to return to the List Manager folder.
7. Select the xwf_email_footer script.
8. Replace the contents of the script with the following:

```
groupId = list_object.listId()
siteId = list_object.siteId
site = getattr(context.Content, siteId)
group_object = getattr(site.groups, groupId)

# we copy the property sheet, because we won't be able to access it
# in the lower layer
group_properties = {}
```

```

for p in group_object.propertyItems():
    group_properties[p[0]] = p[1]
group_properties['id'] = group_object.getId()

try:
    group_properties['division_id'] = site.aq_explicit.getId()
    group_properties['division_title'] = site.aq_explicit.title
except:
    group_properties['division_id'] = ''
    group_properties['division_title'] = ''

# group_properties['canonical_host'] = \
#     group_object.Scripts.get.option('canonicalHost')
divConfig = site.DivisionConfiguration
group_properties['canonical_host'] = divConfig.getProperty('canonicalHost', '')

try:
    from_addr = context.parseaddr(mail.get('from', ''))[1]
except:
    from_addr = ''

if from_addr:
    user = context.acl_users.get_userByEmail(from_addr)
else:
    user = None

files = []
try:
    storage = context.FileLibrary2.get_fileStorage()
    for file_id in file_ids:
        file = storage.get_file(file_id)
        if file:
            files.append(file)
except:
    pass

# Get the virtual file folder "files" from the group.

# Get the public_access_period from "files"
pap = int(getattr(group_object.files, 'public_access_period', 0))
# Turn the public_access_period to a Boolean
pap_set = bool(pap)
# Pass the Boolean to the "email_footer" template
mailto = getValueFor('mailto')

return context.email_footer(REQUEST, list_object=list_object,
                             group_properties=group_properties,
                             getValueFor=getValueFor, title=title,
                             mailto=mailto, mail=mail, body=body,
                             user_object=user, from_addr=from_addr,
                             files=files, post_id=post_id, pap_set=pap_set)

```

9. Click the Save Changes button.

Resources

- Code repository: <https://source.iopen.net/groupserver/>

- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

11.2 Frozen treats

11.2.1 GroupServer 12.05 — Faloodeh Consumed with an Eye on History

Authors Michael JasonSmith; Richard Waid; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2012-05-02

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by OnlineGroups.Net.

Introduction

The *changes to GroupServer* in the Faloodeh release cover sites, groups and profiles — making them all more usable and extensible. You can *get Faloodeh* immediately.

Changes to GroupServer

The changes in Faloodeh are mostly *group improvements*, *profile improvements* and a new *site homepage*. In addition we have made steps towards *easier installation*, and there have been some *minor changes*.

Group Improvements

The most visible change to groups in this release of GroupServer is to the *notifications*. However, the primary improvement is new code to determine if someone *can post* to a group. Underlying this is a system for *more extensible groups*.

Minor changes to groups include the following.

Better Links in Posts: Now <http://youtu.be> links are turned into embedded YouTube videos, and [www.](http://www) “links” are turned into hypertext links. Thanks to [Steven Clift](#) at E-Democracy.Org for sponsoring this improvement.

Rest of Post Button: The button to show the *remainder* of a post (normally just bottom-quoted text) is now labelled “Rest of post”.

Public Access Period: The default public access period for files is now set to 72 hours. This is the period that no password is required to access a file, making it easier for group members who follow conversations using email to access the files that are posted to the group.

Notifications Four notifications have been rewritten. There are HTML versions of both

- The *Welcome* message⁴⁰, which is sent to new group members, and
- The *New Member* notification, which is sent to the administrator when someone joins a group⁴¹.

⁴⁰ A sample *Welcome* message can be viewed at <<http://groupserver.org/groups/development/new-member-msg.html>>

⁴¹ A sample *New Member* message can be viewed at <<http://groupserver.org/groups/development/new-member-admin-msg.html>>

The *Invitation* to join a group has undergone a rewrite too, along with a new default invitation-message from the group-administrator.

Finally, the *Cannot Post* notification has undergone an extensive change, along with the rest of the *can post* system⁴². It now has a HTML variant, as well as a wording change to (hopefully) make it easier to understand.

Can Post The biggest change to groups was a complete rewrite of the system that determines if a user can post. This system is used to

- Send an email to someone who tries to post but is disallowed, and
- To tell someone using the web why he or she cannot post *before* they try to post.

The Can Post system is now self-documenting⁴³, and it allows for *more extensible groups*.

More Extensible Groups Along with the new *can post* code, the definition of what *constitutes* a group has been redefined. GroupServer has always supported three different types of group:

Discussion Group: A group where only group members can post.

Announcement Group: A group where only certain group-members can post.

Support Group: A group where anyone can post, but only group-members can view the posts.

These group-types previously existed as a set of configuration options. Now there are specific marker-interfaces for each of these group-types. Currently only the *can post* system uses them extensively, but other systems will follow in the future.

Profile Improvements

The primary improvements to profiles is a new *Verify Email Address* notification. Like the new *notifications* in the group, the new Verify email message has a HTML version, which is shown by default by most email clients. The wording of the Verify email has also been changed, so it is hopefully easier to understand.

All the notifications with an HTML component are possible because of a new system for sending notifications. This system is documented as part of the `gs.profile.notify` component⁴⁴.

Site Homepage

The site homepage has been completely rewritten, but it looks largely unchanged. Now the page — provided by the new `gs.site.home` product⁴⁵ — can be skinned easily. Extra components can be easily added to it in the future.

Easier Installation

Installation should be more reliable for three reasons. The first is a change of dependencies. The second reason is a more tightly constrained set of software-sources. Finally, there is a new installation script.

Three dependencies have changed.

⁴² A sample *Cannot Post* message can be viewed at <<http://groupserver.org/groups/development/cannot-post.html>>

⁴³ The Can Post system now produces a *Rules List* that shows the rules that are checked whenever someone posts to the group. Example lists can be seen for the [GroupServer Development group](#) and the [GroupServer Announcement Group](#)

⁴⁴ The documentation for how to write a notification is in [the product](#).

⁴⁵ The documentation for how to add components to the new site homepage is in [the site-homepage product](#).

1. The least dramatic is to use [Pillow](#) rather than [PIL](#). The former is a *friendly fork* of the latter, which works with the `easy_install` system that GroupServer uses. This change removes the requirement to download and install both `zlib` and `libjpeg`. The `zlib` dependency in particular was a problem with its frequent changes.
2. The *eGenex mx Extension Series* is now available as an egg, which we now use.
3. Finally, the `PyXML` library is also provided by an egg⁴⁶.

By default the eggs used by the GroupServer installation process now come from only two sources. The eggs that make up GroupServer itself come from <http://eggs.iopen.net/groupserver/base/>. The third-party eggs come from <http://eggs.iopen.net/groupserver/cache/>. This should prove to be more reliable than using the canonical *upstream* servers.

Finally, there is a new installer, the `gs_install_ubuntu.sh` script. For an [Ubuntu](#) system, the new script installs the dependencies, creates the databases in a secure maner, sets up a Python environment, and installs all the GroupServer components.

Minor changes

There have been three sets of minor changes to GroupServer in the Faloodah release: Zope events are better supported, `wsgi` is supported, and there have been numerous coding improvements.

An **event** in Zope is a signal that is raised when an action is carried out⁴⁷. You can write code that subscribes to these events, and takes additional action. (This extra code does not need to alter the original code in any way.) Events are now raised when someone joins a group, joins a site, leaves a group, and leaves a site. This should make these actions more extensible, and allow others to add extra functionality to GroupServer⁴⁸.

WSGI is the method that most Python-based web applications use to communicate to servers. GroupServer can now be served using WSGI, as the last problems with WSGI compatibility have been resolved.

Finally, There have been numerous **coding improvements**. Most have centred around the use of [caching](#). In addition there are new base classes for pages and [viewlets](#) with GroupServer sites, profiles, and groups.

Get Faloodah

To get Faloodah go to [the Downloads page for GroupServer](#) and follow [the GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

The update an existing GroupServer system to Faloodah requires you to first *update the packages*. Then there are two tasks to be carried out in the ZMI: first *update the site marker-interface*, and then the complex task to *update the group marker-interface*

Update the Packages To update the packages carry out the following steps.

1. Download the Faloodah tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball.
3. Copy the file `groupserver-12.05/update-versions.cfg` to your existing GroupServer installation, under the name `versions.cfg`.

⁴⁶ Two other libraries, `libxml2` and `libxslt`, are brought into the system by a small script, which is part of the buildout process. The global libraries *are* used, but they must be symbolically-linked (*symlinked*) because the `virtualenv` system does not link to them.

⁴⁷ The Zope events are provided by [the ZTK documentation](#).

⁴⁸ [The site-member product](#) provides an example of code that both *subscribes* to an event and *raises* an event.

4. Copy the file `groupserver-12.05/update-buildout.cfg` to your existing GroupServer installation, under the name `buildout.cfg`.
5. Copy the file `groupserver-12.05/zope2-2.13.13.cfg` to your existing GroupServer installation.
6. Copy the file `groupserver-12.05/ztk-versions-1.0.6.cfg` to your existing GroupServer installation.
7. In your existing GroupServer installation run:

```
$ ./bin/buildout
```

8. Restart your GroupServer instance.

Update the Site Marker-Interface The *site homepage* improvements require a modification to the sites in the ZMI before they can be seen. To update the sites, carry out the following steps.

1. Log into the ZMI for your GroupServer Installation
2. Navigate to your GroupServer instance in the ZMI (called `groupserver` by default).
3. Add a script by selecting `Script (Python)` from the *Add* menu near the top-right of the ZMI page.
 - Give the new script the Id `add_new_site_homepage`
 - Click the *Add and edit* button.
4. Change the contents of the script to the following:

```
# --mpj17-- The new site home page in gs.site.home works of the single
# IGSSiteFolder marker. As such the old IGSPullPageContentFolder marker
# from the "paragmatic templates" code is not needed. Indeed, it gets
# in the way. This code removes the IGSPullPageContentFolder marker
# interface.
from Products.XWFCore.XWFUtils import remove_marker_interfaces

site_root = context.site_root()
folders = ['Folder', 'Folder (Ordered)']
isSite = lambda d: d.getProperty('is_division', False) and hasattr(d, 'groups')
interfaces = ('Products.GSContent.interfaces.IGSPullPageContentFolder',)
ignore = []

contentFolders = site_root.Content.objectValues(folders)
sites = [ s for s in contentFolders
          if ((s.getId() not in ignore) and
              (s.getProperty('is_division', False)) and
              (hasattr(s, 'groups')))]

print '<html><head><title>Site update</title></head><body>'
for site in sites:
    print '<hr/>'
    print '<p>%s (<code>%s</code>)</p>' % (site.title_or_id(), site.getId())
    remove_marker_interfaces(site, interfaces)
    print '<p>Removed the content marker</p>'
print '</body></html>'
return printed
```

5. Click the *Save Changes* button.
6. Click the *Test* tab. The marker interfaces for all the sites should be updated.

Update the Group Marker-Interface Updating the marker-interfaces in a group is similar to how you *update the site marker-interface*.

1. Log into the ZMI for your GroupServer Installation
2. Navigate to your GroupServer instance in the ZMI (called groupserver by default).
3. Add a script by selecting `Script` (Python) from the *Add* menu near the top-right of the ZMI page.
 - Give the new script the Id `remove_xwf_chat_group_marker`
 - Click the *Add and edit* button.
4. Change the contents of the script to the following:

```
from Products.XWFCore.XWFUtils import remove_marker_interfaces, add_marker_interfaces,

site_root = context.site_root()
folders = ['Folder', 'Folder (Ordered)']
ignore = []

isSite = lambda d: d.getProperty('is_division', False) and hasattr(d, 'groups')

contentFolders = site_root.Content.objectValues(folders)
sites = [ s for s in contentFolders
          if ((s.getId() not in ignore) and isSite(s)) ]

groups = []
isGroup = lambda g: g.getProperty('is_group', False)
for site in sites:
    groupsFolder = getattr(site, 'groups')
    groups += filter(isGroup, groupsFolder.objectValues(folders))

oldInterfaces = ['Products.XWFChat.interfaces.IGSGroupFolder']
newInterfaces = ['gs.group.type.discussion.interfaces.IGSDiscussionGroup']
for group in groups:
    print 'Remove the XWFChat marker from %s (%s)' % (group.title_or_id(), group.getId())
    remove_marker_interfaces(group, oldInterfaces)
    add_marker_interfaces(group, newInterfaces)
return printed
```

5. Click the *Save Changes* button.
6. Click the *Test* tab. The marker interfaces for all the groups should be updated to *discussion* groups.
7. Change the *support* groups to the correct marker-interface by carrying out the following for each group.
 - (a) Navigate to the group.
 - (b) Click on the *Interfaces* tab.
 - (c) Select `gs.group.type.discussion.interfaces.IGSDiscussionGroup` in the *Provided Interfaces* list.
 - (d) Click the *Remove* button. The discussion-group marker interface will be removed.
 - (e) Select `gs.group.type.support.interfaces.IGSSupportGroup` in the *Available Marker Interfaces* list.
 - (f) Click the *Add* button. The support-group marker interface will be added.
8. Change the *announcement* groups to the correct marker-interface by carrying out the following for each group.
 - (a) Navigate to the group.

- (b) Click on the *Interfaces* tab.
 - (c) Select `gs.group.type.discussion.interfaces.IGSDiscussionGroup` in the *Provided Interfaces* list.
 - (d) Click the *Remove* button. The discussion-group marker interface will be removed.
 - (e) Select `gs.group.type.announcement.interfaces.IGSAnnouncementGroup` in the *Available Marker Interfaces* list.
 - (f) Click the *Add* button. The announcement-group marker interface will be added.
9. Finally, remove the legacy group-support from your site:
- (a) Copy the file `groupserver-12.05/versions.cfg` to your existing GroupServer installation, under the name `versions.cfg`.
 - (b) Copy the file `groupserver-12.05/buildout.cfg` to your existing GroupServer installation, under the name `buildout.cfg`.
 - (c) In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

- (d) Restart your GroupServer instance.

11.2.2 GroupServer 11.08 — Banana Split Eaten in a Comfortable Silence

Authors [Michael JasonSmith](#); [Richard Waid](#); [Dan Randow](#)

Contact [Michael JasonSmith](#) <mpj17@onlinegroups.net>

Date 2011-08-31

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

Introduction

The main update that is present in the Banana release of GroupServer is a *Facebook signup* system. You can *get Banana Split* immediately.

Acknowledgements

Thanks to [Steven Clift](#) for his support for the *Facebook signup* system.

Facebook Signup

Previously, new members had to provide all their profile information during signup. Now new members have the option of logging into [Facebook](#) to retrieve the profile information. Signing up using Facebook is beneficial in two ways.

1. The email address of the new member does not have to be verified, as Facebook has already verified the address. This reduces the number of steps required for sign-up from three to two.
2. Much of the profile information does not have to be added, as it is copied in from Facebook.

The new member has a choice to use the older signup system that just uses email, or the new Facebook signup. The latter option is automatically available after the administrator has *setup* Facebook signup.

Setup

You must acquire two values from Facebook to enable Facebook Signup.

Application ID Value This is the identifier for your Facebook *application*.

Application Secret Value This is the secret key that is passed from your site to Facebook.

To get these values you will need to register your site as an *application* with Facebook. Do this by visiting <https://developers.facebook.com/> and following the instructions. Facebook will then supply you with an App ID and App Secret. Once you have these you can carry out the following tasks.

1. Log into the PostgreSQL database used GroupServer by using the following command:

```
$ psql -Upgsql_user pgsql_dbname
```

Note that `pgsql_user` and `pgsql_dbname` are the database user and database name that were set during installation. Both can be found in the `instance.cfg` file in the installation directory of GroupServer.

2. Run the following SQL to add the App ID and App Secret to the options table:

```
INSERT INTO option
VALUES (
    'gs.profile.signup.facebook',
    'app_id',
    NULL,
    NULL,
    App ID
);

INSERT INTO option
VALUES (
    'gs.profile.signup.facebook',
    'app_secret',
    NULL,
    NULL,
    App Secret
);
```

Note the App ID and App Secret in the above SQL are the values provided by Facebook. Both values will need to be in quotes.

Once the data is in the `options` table the Sign Up page will automatically show the option to sign up using Facebook.

Get Banana Split

To get Banana Split go to the [Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To update an existing GroupServer system to Banana Split carry out the following steps.

1. Download the Banana Split tar-ball from [the GroupServer download page](#).

2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.08/versions.cfg` to your existing GroupServer installation.
4. Copy the file `groupserver-11.08/buildout.cfg` to your existing GroupServer installation.
5. In your existing GroupServer installation run:

```
$ ./bin/buildout
```

11.2.3 GroupServer 11.07 — Frozen Yoghurt Accompanied by Carols

Authors [Michael JasonSmith](#); [Richard Waid](#); [Dan Randow](#)

Contact [Michael JasonSmith](#) <mpj17@onlinegroups.net>

Date 2011-07-28

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

Introduction

The main update that is present in the Frozen Yoghurt release of GroupServer is a *new stylesheet* for devices with narrow displays . There is also a new converter for *HTML-formatted messages*. In addition a number of *minor fixes* are part of this release. You can *get Frozen Yoghurt* immediately. Work is now starting on GroupServer 11.08 — Banana Split Eaten in a Comfortable Silence.

Acknowledgements

Thanks to [Steven Clift](#) for his support with the *new stylesheet*. Thanks also to [Le Coyote](#) and [Alice Rose](#) for testing an early release of Frozen Yoghurt.

New Stylesheet

Devices with narrow displays, such as mobile phones and tablets, now have a dedicated stylesheet ⁴⁹. The system uses [CSS3 Media Queries](#) to reformat the page when the browser window is narrow. Pages that have been specifically optimised for small-screen devices include

- The site homepage,
- The group page,
- The topic page and
- The image page.

Many other pages work on small screens, including the post page and the topics page.

⁴⁹ The new mobile CSS closes [Ticket 444](#). More information on the mobile stylesheet can be found in [the topic in GroupServer Development](#).

HTML-Formatted Messages

Email messages can be in multiple formats, including plain text and HTML. Sometimes email messages are formatted using just HTML. When this occurs GroupServer converts the HTML version of the message into a plain-text version. It is this plain-text version of the post that is displayed on the Web pages. The code that converts the HTML-message into plain-text has been updated ⁵⁰. The updated code uses the [standard Python HTML Parser library](#) to parse the message, and also handles URLs better.

Minor Fixes

Minor fixes in Frozen Yoghurt include the following.

- The position of the Sticky form in a topic has been moved, so it is visible when there is only one post ⁵¹.
- Files that are part of hidden posts are hidden from the index of files at the top of the topic ⁵². (Files that were part of hidden posts were never able to be viewed.)
- A new configuration system for GroupServer is present, but not used. The `gs.option` product is part of an ongoing effort to move more data out of the ZODB and into the [PostgreSQL](#) relational database. The `README` in the base of the `gs.option` product has more details.
- An error with the Can Post code was fixed.
- The word *Home* has been dropped from the group page, as it added nothing and took up valuable space (which was especially noticeable with the [new stylesheet](#)).

Get Frozen Yoghurt

To get Frozen Yoghurt go to the [Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#). Those who already have a functioning installation can [update an existing GroupServer system](#).

Update an Existing GroupServer System

To update an existing GroupServer system to Frozen Yoghurt you must [update the database](#) and then [update the packages](#).

Update the Database The new `options` code requires the creation of a new table to record information about the properties of the various parts of GroupServer (see [Minor Fixes](#)). To update the database carry out the following steps.

1. Log into the PostgreSQL database used GroupServer by using the following command:

```
$ psql -Upgsql_user pgsql_dbname
```

Note that `pgsql_user` and `pgsql_dbname` are the database user and database name that were set during installation. Both can be found in the `instance.cfg` file in the installation directory of GroupServer.

2. Run the following SQL to create the `options` table:

```
CREATE TABLE option (  
    component_id    TEXT    NOT NULL,  
    option_id       TEXT    NOT NULL,  
    site_id         TEXT    NOT NULL,
```

⁵⁰ Updating the converter closes [Ticket 596](#).

⁵¹ Fixing the position of the Sticky form in a topic closes [Ticket 705](#).

⁵² Hiding posts that are part of hidden files closes [Ticket 692](#).


```

        group_id      TEXT NOT NULL,
        value         TEXT
    );

    CREATE UNIQUE INDEX option_idx
    ON option (component_id, option_id, site_id, group_id);

```

Update the Packages To upgrade the pages in an existing GroupServer system to Frozen Yoghurt carry out the following steps.

1. Download the Frozen Yoghurt tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.07/versions.cfg` to your existing GroupServer installation.
4. Copy the file `groupserver-11.07/buildout.cfg` to your existing GroupServer installation.
5. In your existing GroupServer installation run:

```

$ ./bin/easy_install -U setuptools && ./bin/python bootstrap.py
$ ./bin/buildout

```

The `setuptools` have to be updated, and `bootstrap.py` rerun, because the version of Zope used by GroupServer has undergone an update.

11.2.4 GroupServer 11.06 — Soft Serve from Mr Whippy

Authors [Michael JasonSmith](#); [Richard Waid](#); [Dan Randow](#)

Contact [Michael JasonSmith](#) <mpj17@onlinegroups.net>

Date 2011-07-04

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

Introduction

The two main updates to GroupServer that are present in the Soft Serve release are an update to the code that determines if someone *can post*, and a rebuild of the *site administration pages*. In addition there are other *minor fixes* that have been made. You can *get Soft Serve* immediately. Work is now starting on GroupServer 11.07 — Frozen Yoghurt Accompanied by Carols.

Acknowledgements

Thanks to [Le Coyote](#) and [DJS](#) for their very helpful suggestions, and testing early releases of Soft Serve.

Can Post

When someone posts to a group the system checks to see if the user has the correct permissions. The code that does this is known as the *can post* code. In Soft Serve this code has undergone two main changes.

1. Email addresses can be blocked from posting ⁵³. This mechanism, known as blacklisting, extends the existing system that prevented addresses on the black-list from receiving posts. There is no user-interface for this system.
2. The code has been moved to its own product ⁵⁴.

Site Administration Pages

The site-administration pages did not work well. All the site administration pages have been updated ⁵⁵. In particular, the pages that allow an administrator to set the site timezone ⁵⁶ and change the site name ⁵⁷ have been fixed so they work correctly.

Minor Fixes

Minor fixes in Soft Serve include the following.

- If a member with a single unverified address resets a password then that address will become verified ⁵⁸.
- The `Reply-to` address of an invitation is now set to the address of the administrator who issued the invitation ⁵⁹.
- Only posting members of an announcement group are shown a link to start a topic ⁶⁰.
- The name of the initial GroupServer instance and GroupServer site have been changed ⁶¹.
- A coding error with the Members page has been fixed ⁶².

Get Soft Serve

To get Soft Serve go to the [Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#). Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To upgrade an existing GroupServer system to Soft Serve you must first *update the packages* and then *update the ZMI*.

⁵³ Adding the code for blocking a post closed [Ticket 459](#). This enhancement mostly helps Support groups, which otherwise lack a way to prevent people from posting.

⁵⁴ There is a long-running project to move code from the large `Products` eggs to many more smaller `gs` eggs. The move of the Can Post code to `gs.group.member.canpost` from the old `Products.GSGroupMember` is part of this. Moving the code closes [Ticket 423](#).

⁵⁵ Updating the site administration pages closes [Ticket 620](#).

⁵⁶ Updating the site-timezone page closes [Ticket 662](#).

⁵⁷ Updating the page that allows a site-name to be changed closes [Ticket 607](#).

⁵⁸ Only the simple case of a single-address being verified is currently handled. Verifying an email address when the password is reset closes [Ticket 480](#).

⁵⁹ Prior to Soft Serve, the `Reply-to` address was set to the email address of the site support. Setting the `Reply-to` to the address of the administrator that issued the invitation closes [Ticket 681](#).

⁶⁰ In an announcement group there is a distinction between group members that post (*posting members*) and members that just view posts (*normal members*). Prior to Soft Serve all group members saw the link on the homepage to the *Start a Topic* page. A normal member would see an error if he or she followed that link. Just showing the link on the homepage to the posting members closes [Ticket 530](#).

⁶¹ Prior to Soft Serve the initial GroupServer instance, site and group had names that contained `example`. Now the instance is called `groupserver`, the initial site is called `initial_site`, and the initial group is called `example_group`. Renaming the instance and site [Ticket 690](#).

⁶² Fixing the coding error on the Members page closes [Ticket 680](#).

Update the Packages Carry out the following steps to update the package versions.

1. Download the Soft Serve tar-ball from [the GroupServer download page](#).
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.06/versions.cfg` to your existing GroupServer installation.
4. Copy the file `groupserver-11.06/buildout.cfg` to your existing GroupServer installation.
5. In your existing GroupServer installation run:

```
$ ./bin/buildout
```

Update the ZMI The update to the *site administration pages* requires an instance to be deleted from the Zope Management Interface (ZMI) before it will work. To delete the instance carry out the following tasks.

1. Start GroupServer:

```
$ ./bin/instance fg
```

2. View the ZMI. If you did not change the defaults the ZMI can be accessed from [<http://localhost:8080/manage/>](http://localhost:8080/manage/).
3. Enter the user-name and password of the ZMI administrator.
4. Visit the ZMI page for the Example Site. This should be under `example`, `Content`, `example_site`.
5. Select the check-box next to `admindivision` (`Administer Site`).
6. Click the Delete button. The `admindivision` instance should be deleted.

After deleting the `admindivision` instance you should be able to view the new site administration pages. To view these pages you must view the homepage for your site and click on the link to administer the site.

Footnotes

11.2.5 GroupServer 11.05 — Eskimo Pie with Middle Class Guilt

Authors Michael JasonSmith; Richard Waid; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2011-05-27

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by OnlineGroups.Net.

The main update to GroupServer that is present in the Eskimo Pie release is the new *request membership* system. However there are other *minor fixes* that have been made. You can *get Eskimo Pie* immediately. Work is now starting on GroupServer 11.06 — Soft Serve from Mr Whippy.

Request Membership

The Request Membership system⁶³ allows someone to request membership of a private group. The group administrator is informed of the request by email. The administrator can then accept or decline the request using a Web interface.

⁶³ Adding the Request Membership system (finally) closes [Ticket 49](#). More information in the Request Membership system can be found in the [Request Membership topic](#).

Effectively the Request Membership system works like the system that is used to invite people to join groups, but in reverse.

Minor Fixes

Minor fixes in Eskimo Pie include the following.

- An update to the Start a Group system so the system administrator is made the group administrator ⁶⁴.
- When an email address is verified it is made the preferred delivery address if the user has no other preferred delivery addresses ⁶⁵.
- The list of files that are associated with a post are more clearly demarcated from the body of the message ⁶⁶.
- Long URLs no longer word-wrap in the body of posts ⁶⁷.
- Zope has been updated to 2.3.6 ⁶⁸. This *may* cause problems with some configurations, as IPv6 support is now present.

Get Eskimo Pie

To get Eskimo Pie go to [the Downloads page for GroupServer](#) and follow [the GroupServer Installation documentation](#).

Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To upgrade an existing GroupServer system to Eskimo Pie you must first *update the database* and then *update the packages*.

Update the Database The *Request Membership* update requires the creation of a new table to record information about membership requests. To update the database carry out the following steps.

1. Log into the PostgreSQL database used GroupServer by using the following command:

```
$ psql -Upgsql_user pgsql_dbname
```

Note that `pgsql_user` and `pgsql_dbname` are the database user and database name that was setup during installation. Both can be found in the `instance.cfg` file in the installation directory of GroupServer.

2. Run the following SQL to create the request membership table:

```
CREATE TABLE user_group_member_request (  
    request_id          TEXT          PRIMARY KEY,  
    user_id             TEXT          NOT NULL,  
    site_id             TEXT          NOT NULL,  
    group_id            TEXT          NOT NULL,  
    request_date         TIMESTAMP WITH TIME ZONE NOT NULL,  
    message              TEXT,  
    responding_user_id   TEXT,  
    response_date        TIMESTAMP WITH TIME ZONE,
```

⁶⁴ Setting the group administrator when the group is started closes [Ticket 611](#).

⁶⁵ Setting the verified address as the preferred delivery address closes [Ticket 661](#).

⁶⁶ Enhancing the file notification area closes [Ticket 664](#)

⁶⁷ Fixing the URLs so the do not break over multiple lines closes [Ticket 671](#).

⁶⁸ Updating to Zope 2.3.6 is something I am sure to regret, because it has the potential to make GroupServer installation even more tricky than before. However, it is done, and I have closed [Ticket 672](#).

```

        accepted          BOOLEAN
    );

```

Update the Packages Carry out the following steps to update the package versions.

1. Download the Eskimo Pie tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.05/versions.cfg` to your existing GroupServer installation.
4. Copy the file `groupserver-11.05/buildout.cfg` to your existing GroupServer installation.
5. In your existing GroupServer installation run:

```
$ ./bin/buildout
```

11.2.6 GroupServer 11.04 — Slushy Followed by a Pounding Headache

Authors Michael JasonSmith; Richard Waid; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2011-04-29

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by OnlineGroups.Net.

The Slushy release of GroupServer contains two main *enhancements*. You can *get Slushy* immediately. Work is now starting on GroupServer 11.05 — Eskimo Pie with Middle-Class Guilt.

Enhancements

The two main enhancements to GroupServer are

- Posts can now be hidden (see *Hide a Post* below) and
- *Full email addresses* can now be used to issue invitations.

There are also some *minor fixes*.

Hide a Post

The most significant change to GroupServer is the introduction of an interface to allow posts to be hidden⁶⁹. Both administrators and the author of the post can hide it. People trying to view the post, by itself or in a topic, will see the reason the post was hidden instead of the message. Anyone trying to view a file or image that was added as part of the post will see an error-page that states why the post was hidden.

⁶⁹ The new Hide a Post interface closes [Ticket 316](#) and many minor tickets that were related to hiding a post.

Full Email Addresses

Prior to Slushy GroupServer would only accept the portion of an email address with an @ in it — such as `support@onlinegroups.net`. Now the *Invite a New Member* page can handle the full address. This address can include a name, such as "OnlineGroups.Net Support" `<support@onlinegroups.net>`⁷⁰. Other pages will be converted to support full addresses over time.

Minor Fixes

Minor fixes in Slushy include

- A fix to the *Join and Leave Log* so it now uses jQuery UI tabs⁷¹,
- A fix to the *view* link in the content editor so it works⁷², and
- Resolving an error with the *Change Email Settings* page that occurred when someone had no preferred email address set⁷³.

Get Slushy

To get Slushy go to [the Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#).

Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

To upgrade an existing GroupServer system to Slushy you must first *update the database* and then *update the packages*.

Update the Database The *hide a post* update requires a change to the database tables that are used to record information about topics and posts. It also requires the creation of a new table to record information about why posts have been hidden.

To update the database carry out the following steps.

1. Log into the PostgreSQL database used GroupServer by using the following command:

```
$ psql -Upgsql_user pgsql_dbname
```

Note that `pgsql_user` and `pgsql_dbname` are the database user and database name that was setup during installation. Both can be found in the `instance.cfg` file in the installation directory of GroupServer.

2. Run the following SQL to update the post and topic tables:

```
ALTER TABLE topic ADD COLUMN hidden TIMESTAMP WITH TIME ZONE;  
ALTER TABLE post ADD COLUMN hidden TIMESTAMP WITH TIME ZONE;
```

3. Finally, create the hidden post table by running the following SQL commands:

⁷⁰ Support a full email address closes [Ticket 445](#).

⁷¹ Switching the *Join and Leave Log* to use tabs from jQuery.UI closes [Ticket 641](#).

⁷² Fixing the issue with the *view* link closes [Ticket 642](#).

⁷³ Fixing the coding error on with the *Change Email Settings* page closes [Ticket 660](#)

```
CREATE TABLE hidden_post (
  post_id      TEXT REFERENCES post ON UPDATE CASCADE,
  date_hidden  TIMESTAMP WITH TIME ZONE NOT NULL,
  hiding_user  TEXT NOT NULL,
  reason       TEXT
);
CREATE UNIQUE INDEX hidden_post_pkey
  ON hidden_post
  USING BTREE(post_id, date_hidden);
```

Update the Packages Carry out the following steps to update the package versions.

1. Download the Pineapple Snow tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.04/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

11.2.7 GroupServer 11.03 — Pineapple Snow at a Child's Party

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2011-03-29

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The Pineapple Snow release of GroupServer contains four main enhancements and corrections.

- There is an *Updated Change Email Settings Page*.
- Searching will now search the topic and post titles ⁷⁴.
- An *Edit this page* link is sometimes shown to logged out users on editable pages ⁷⁵.
- The *Accept Invitation* page looks a lot better ⁷⁶.

You can *get Pineapple Snow* immediately. Work is now starting on GroupServer 11.04 — Slushy followed by a Pounding Headache.

Updated Change Email Settings Page

The Change Email Settings page has had an extensive redesign ⁷⁷. Email addresses can be dragged from one list to another in order to characterise them. The old interface relied on menus and multiple buttons, and was far harder to use. In addition the new interface automatically updates when an email address is verified ⁷⁸.

⁷⁴ Updating search closes [Ticket 227](#)

⁷⁵ The *Edit this page* link is only shown to the Anonymous user if logged in members can edit the page. Showing this link closes [Ticket 384](#)

⁷⁶ The *Accept Invitation* was hit by a CSS error. Fixing the error closes [Ticket 618](#) and fixes a related error on other pages with radio buttons and check buttons.

⁷⁷ The redesign of the *Change Email Settings* page closes [Ticket 478](#) and [Ticket 472](#). The topic in [GroupServer Development](#) details the new design.

⁷⁸ Automatically updating the *Change Email Settings* page when an address is verified closes [Ticket 276](#).

The new interface uses the jQuery UI JavaScript library, just like the group homepage. jQuery UI is an excellent library builds on the already great jQuery library ⁷⁹.

Get Pineapple Snow

To get Pineapple Snow go to [the Downloads page for GroupServer](#) and follow the [GroupServer Installation documentation](#).

Those who already have a functioning installation can *update an existing GroupServer system*.

Update an Existing GroupServer System

Carry out the following steps to update the package versions.

1. Download the Pineapple Snow tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.03/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

11.2.8 GroupServer 11.02 — Tartufo Nibbled in Polite Company

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2011-02-28

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The Tartufo release of GroupServer mostly consists of enhancements so errors are handled better.

- A new *Unexpected Error* page provides more detail to the user. In addition, the pre-written message to support contains a full stack-trace. This information can be used to find and eliminate the error ⁸⁰.
- Many errors were generated when non-existent pages were accessed. These errors have been fixed, primarily in the mailing-list code. This should make problems recorded in the error-log maintained by Zope easier to find, as there should be fewer items in the log ⁸¹.
- Changes to email addresses are now better recorded ⁸². This should help diagnose problems better if there is an error subsequent to an address changing.
- The jQuery code shipped with GroupServer has been updated to the latest stable version (1.4.4).
- The code that generates the image, post, and topic pages has been refactored. This should make those pages easier to maintain.

Work is now starting on GroupServer 11.03 — Pineapple Snow at a Child's Party.

⁷⁹ The new *Change Email Settings* page uses the [sortable widgets](#) and the [dialog widget](#) from jQuery UI.

⁸⁰ The new Unexpected Error page is detailed in a [post to GroupServer Development](#).

⁸¹ The *traversal* errors were those caused by not handling errors that were thrown by pages in the messages, profile, and files area.

⁸² Recording more information about email addresses changing closes [Ticket 432](#).

Update an Existing GroupServer System

Carry out the following steps to update the package versions.

1. Download the Tartufo tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.02/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

11.2.9 GroupServer 11.01 — Baked Alaska with and Eye on the Soviets

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2011-01-31

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The Baked Alaska release of GroupServer mostly consists of changes to the email verification system.

- Error handling is improved. Following an old verification link will show a suggestion to view your profile, rather than an Unexpected Error page. In addition the corner cases of a missing verification ID and broken ID are handled.
- The Verification page now more closely conforms to Web standards.
- The verification code (`gs.profile.email.verify`) is now easier to understand and extend.

In addition the site homepage has been modified. Now the groups and profile image of a logged in user are shown. This should make groups easier to find.

Work is now starting on GroupServer 11.02 — Tartufo Nibbled in Polite Company.

Update an Existing GroupServer System

Carry out the following steps to update the package versions.

1. Download the Baked Alaska tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-11.01/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

11.2.10 GroupServer 10.12 — Lemon Ice in the Cool of the Evening

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2010-12-17

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](http://creativecommons.org/licenses/by-sa/3.0/) by OnlineGroups.Net.

The Lemon Ice release of GroupServer mostly consists of changes to the CSS.

- New layout, based on the layout of OnlineGroups.net.
- Two new color schemes (skins):
 1. Blue, based on [the E-Democracy.org](http://theE-Democracy.org) Forums
 2. Green, based on OnlineGroups.Net
- *Fit and finish* work to improve the look of GroupServer.
- Integration of the jQuery.UI code into the default CSS.

Work is now starting on GroupServer 11.01 — Baked Alaska with an Eye on the Soviets.

Update an Existing GroupServer System

To update an existing GroupServer system to Lemon Ice you will have to *update the package versions*. To use the new skins you will have to *update the virtual hosting*.

Update the Package Versions

Carry out the following steps to update the package versions.

1. Download the Lemon Ice tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-10.12/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

Update the Virtual Hosting

The virtual hosting will need to be updated in order to use the new skins. Without the new skins the default black-and-white color scheme will be shown. By default Zope is set up to provide the virtual hosting. To change the Zope setup carry out the following

1. Log into the ZMI for your site.
2. Select `virtual_hosting`. You should see the page for the Virtual Host Monster.
3. Select the `Mappings` tab. You should see a line that starts with the name of your site (`gctest` by default) that is followed path to the site (`/example/Content/example_site/`).
4. Add `++skin++gs_blue` to the end of the path for the blue skin. Alternatively add `++skin++gs_green` for the green skin.
5. Click the *Save Changes* button.
6. Visit the homepage your site (`<http://gctest/>` by default).

7. Reload the page. You may need to hold down `Shift` when clicking the Reload button to ensure you reload the CSS.

11.2.11 GroupServer 10.11 — Kulfi Craved while Caving

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2010-11-30

Organization GroupServer.org

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The changes below form the bulk of what is new in the Kulfi release of GroupServer:

- Enhancements to groups, including a clearer *group properties page*, a more flexible *delivery settings page*, a simplified *start a group* process, and a new *group homepage about tab*.
- *Management of a group member* is now easier.
- There have been some *set password* enhancements and *notification updates*.
- Finally, the *buildout* process has been improved.

All these enhancements will be available to new GroupServer installations, and those who *update an existing groupserver system*.

Work is now progressing on GroupServer 10.12 — Lemon Ice in the Cool of the Evening.

Group Properties Page

The page used to set the group properties has been simplified ⁸³, following a discussion about the properties ⁸⁴. There are fewer properties, as many properties have been replaced by the *group homepage about tab*. The properties that remain have been given clearer names.

The new group properties page is now a standard `zope.formlib` form, rather than relying on the older GroupServer `XForms` code. This means that standard features of forms on GroupServer are present, such as popup help.

Delivery Settings Page

The page used to change the email delivery settings of a group member has been updated ⁸⁵. The biggest advantage to the new page is that administrators can now change the delivery settings of group members. For members the page looks and behaves much as it did before. However, the system has undergone a major rework behind the scenes. This will make the page easier to maintain, extend and change.

Start a Group

The start a group system has been simplified to a one-step process ⁸⁶. Rather than a separate Preview step the new system dynamically previews the URL and email address of the new group. It also dynamically checks if the group identifier is unique.

⁸³ The new group properties page closes [Ticket 292](#)

⁸⁴ The GroupServer Development group hosted a lively debate about topics

⁸⁵ Creating a new email-settings page closes [Ticket 371](#)

⁸⁶ Simplifying the process used to start a group closes [Ticket 304](#)

It is not possible to create *announcement* groups with the simplified system. However, announcement groups are still supported by GroupServer and will continue to be supported. To create an announcement group first create a discussion group. Then use the ZMI to change the `group_template` property to `announcement`.

Group Homepage About Tab

The new group homepage about tab replaces the little-used Charter page⁸⁷. The new tab provides a blank area where a site administrator is free to write any HTML. The WYMeditor is used to edit the contents of the tab⁸⁸.

Management of a Group Member

A group administrator is now shown a *Manage Member* link when viewing the profile of a member⁸⁹. This should make managing members even easier than before.

Set Password

The code used to set and reset passwords has been updated. Members will be shown better error messages when they follow password-reset links multiple times⁹⁰. In addition all the Set Password pages have been updated to use a single text entry, rather than two password entries⁹¹.

Notification Updates

Some notifications that are stored in the ZMI have been updated. In addition better feedback is given for those who fail to change the group-delivery settings by email⁹².

Buildout

Buildout is the system that GroupServer uses to install the system. The buildout process has been improved in three ways. First, some issues with the environment relating to installing `lxml` have been resolved⁹³. Second, the standard system is used to add the default administrator and user to the example group. This should reduce the chance of errors occurring in the future. Finally, summary information about the new GroupServer site is displayed at the end of the buildout process. This should make it easier to start and view the new GroupServer site.

Update an Existing GroupServer System

To update an existing GroupServer system to Kulfi you will have to *update the package versions*, *update the SQL*, and *update the ZODB*.

⁸⁷ Creating the About Tab closes [Ticket 493](#)

⁸⁸ GroupServer uses the excellent [WYMeditor](#) as its HTML editor. Pages that use the editor include *Change Profile*, *Change Site Introduction* and all pages that are editable with the Content Manager (such as *About* and *Policies*).

⁸⁹ Creating a link from the profile page to the manage member page closes [Ticket 515](#)

⁹⁰ Knowing when a password has been reset closes [Ticket 326](#)

⁹¹ Why text entries are used to set passwords is explained in this [blog post](#)

⁹² The clean up to notifications closes three tickets:

1. [Ticket 205](#)
2. [Ticket 231](#)
3. [Ticket 531](#)

⁹³ The `lxml` improvements should resolve the issues that Tom had when installing GroupServer

Update the Package Versions

Carry out the following steps to update the package versions.

1. Download the Kulfi tar-ball from [the GroupServer download page](#)
2. Uncompress the tar-ball.
3. Copy the file `groupserver-10.11/versions.cfg` to your existing GroupServer installation.
4. In your existing GroupServer installation run:

```
$ ./bin/buildout -N
```

Update the SQL

The *set password* enhancements require an update to the relational database that stores GroupServer data. Carry out the following steps to update the database.

1. In `instance.cfg` look up `pgsql_dbname` and the `pgsql_user`, and note their values.
2. Run the following command:

```
$ psql -U {pgsql_user} {pgsql_dbname} -c "ALTER TABLE "\
    "password_reset ADD COLUMN reset TIMESTAMP WITH TIME ZONE "\
    "DEFAULT NULL;"
```

Older installations will also have to update the table used to record invitations:

```
$ psql -U {pgsql_user} {pgsql_dbname} -c "ALTER TABLE "\
    "user_group_member_invitation ADD COLUMN initial_invite "\
    "BOOLEAN DEFAULT FALSE, ADD COLUMN withdrawn_date TIMESTAMP "\
    "WITH TIME ZONE, ADD COLUMN withdrawing_user_id TEXT;"
```

Update the ZODB

To get the *notification updates* into an existing GroupServer system you will have to update the email-templates in the ZODB. Email support@onlinegroups.net or [GroupServer Development](#) if you need a hand with this.

11.2.12 GroupServer 10.10 — Gelato while Viewing the Sights

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2010-10-26

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The changes below form the bulk of what is new in the Gelato release of GroupServer. Work is now progressing on GroupServer 10.11 — Kulfi Craved while Caving.

New Group Homepage

The homepage used by groups has had a major redesign ⁹⁴. Now it is divided into multiple tabs. This allows for information to be better organized, and more easily found. Overall the look is cleaner, and the page is easier to use.

Better Bouncing

Sometimes an email to a group member will not get through, and the message bounces. GroupServer has always recorded these bounces. Now this information is presented to the group administrator, who can see a log of failed attempts to send the member an email message ⁹⁵.

Resend Invitation

The group administrator now has the option of resending an invitation that did not get through to the new group member ⁹⁶. There have been many other minor improvements to the invitation system thanks to the feedback from the users of [OnlineGroups.Net](#).

Notification Updates

The email notifications that are sent out by GroupServer in response to an event have been given a through update. They are now clearer and contain more useful information than before ⁹⁷.

New Favicon

GroupServer now has a new multi-resolution favicon, which is shown in the location bar of the browser ⁹⁸.

New Help System

The user manual and administration manuals are out of date and in need of work. To help with this a new help system has been introduced ⁹⁹. It allows for sections of the manuals to be more easily maintained, which will hopefully allow for better manuals in the future.

⁹⁴ The details of the new group homepage is available in the [Group Homepage Rebuild](#) topic in [GroupServer Development](#). Deploying the new homepage closes [Ticket 426](#) and a slew of sub-tickets.

⁹⁵ Showing a log of the failed attempts to send a message closes

- [Ticket 15!](#)
- [Ticket 509](#)

⁹⁶ Allowing administrators to reissue an invitation closes [Ticket 489](#)

⁹⁷ Updating the notifications closes eight tickets

- [Ticket 163](#)
- [Ticket 215](#)
- [Ticket 216](#)
- [Ticket 221](#)
- [Ticket 222](#)
- [Ticket 223](#)
- [Ticket 466](#)
- [Ticket 513](#)

⁹⁸ Creating the new favicon for GroupServer closes [Ticket 510](#)

⁹⁹ Creating the new help pages closes [Ticket 485](#)

Switch to Zope 2.13 Beta

GroupServer now runs on the the latest version of [Zope 2](#). The new features in Zope are used for the *new group homepage*, and the *new help system*.

11.2.13 GroupServer 10.09 — Spaghettieis with a Wafer of Confusion

Authors Michael JasonSmith; Richard Waid; Alice Rose; Dan Randow

Contact Michael JasonSmith <mpj17@onlinegroups.net>

Date 2010-09-09

Organization [GroupServer.org](#)

Copyright This document is licensed under a [Creative Commons Attribution-Share Alike 3.0 New Zealand License](#) by [OnlineGroups.Net](#).

The changes below form the bulk of what is new in the Spaghettieis release of GroupServer. Work is now progressing on GroupServer 10.10 — Gelato while Viewing the Sights.

Invitations

All the pages that deal with issuing and responding to invitations have changed.

Issuing a Single Invitation

The biggest visible change with invitations is to the page used to issue a single invitation. The administrator can write a **personal message** to the invited member. The administrator can preview the message, and preview the response page¹⁰⁰. Hopefully this will encourage more invitations to be written, as they are less likely to get marked as spam, and are more likely to get a response.

The **from address** of the invitation is now set to the address of the administrator¹⁰¹. This will hopefully reduce the chances of the invitation being caught in a spam filter even further.

The **email address** entry on the invitation page handles incorrect data better, so only email addresses can be entered¹⁰². In the future I would like GroupServer to be more flexible with the format of the addresses that are entered, but today is not that day¹⁰³.

Inviting Site Members

The page for inviting site members has been spruced up, but it has not been overhauled in any major way. The **invitation message** can now be changed. However, it cannot be previewed because I am lazy and did not want to go through the hassle of writing a preview of the Accept Invitation page for existing site members. If someone actually uses Invite Site Members and bothers to write a ticket I may get some motivation to write a preview page.

¹⁰⁰ The [GroupServer Development](#) online group contains examples of

- The invitation-message preview and
- The response page.

¹⁰¹ Setting the `From` address in invitations correctly will close [Ticket 290](#).

¹⁰² Being strict about what can be entered as an email address fixes [Ticket 325](#). The fix also corrects the same error with the sign up page.

¹⁰³ More tolerant email address handling will hopefully come in [Baked Alaska](#).

Inviting People In Bulk

The system for inviting people in bulk, by uploading a **CSV**, remains much the same. Those who use the page a lot will notice some better handling of required profile-attributes, but that is about it. The **invitation message** cannot be changed when inviting people in bulk ¹⁰⁴.

New Members Accepting Invitations

The page used to accept invitations has undergone a major rewrite. The mechanics (enter a password and join) stay the same. However, the biography of the administrator who issued the invitation is now shown, as well as some sample statistics about the group ¹⁰⁵.

New members are now sent a **welcome message** ¹⁰⁶. Hopefully new members will refer to the Welcome message when trying to get back to the group homepage, rather than relying on the invitation. However, if the new member does follow the invitation again he or she will be shown a page that says that the invitation has been accepted, and to the group homepage and *bookmark* it.

If an invitation is **declined** it is noted by the system ¹⁰⁷. Sadly the unwanted profile is not deleted, due to an odd issue with how Zope redirects. I plan to implement a system for removing the unwanted profiles, but it will have to be a manual process for now ¹⁰⁸.

All **group administrators** will now get an email saying that a new member has joined the group ¹⁰⁹. If someone declines an invitation then only the person who issued the invitation is informed.

Existing Members Accepting Invitations

The **profile page** now has a list of invitations on it ¹¹⁰. Clicking on the invitation will go to the *Accept Invitations* page.

The page that allows an existing site-member to accept or decline invitations stays much as it was. It does gain the **group summary** from the *Accept Invitation* page that is shown to new members. It also shares the fixes to the notifications that are sent to the new member and the group administrators.

Join and Leave

The pages that are by people to join a group without an invitation have also changed.

Join and Sign Up

The *Join* and *Sign Up* pages superficially look exactly the same. However, the code for joining a group is shared with the two *Accept Invitations* pages, so the fixes to notifications are also shared. The only specific improvement is a fix to prevent administrators who join **moderated groups** from being moderated ¹¹¹.

¹⁰⁴ OnlineGroups.Net could not ship an *Invite by CSV* page with an editable message as it is an invitation to spam people. However, we are not be averse to someone writing a page with an editable message and including that in GroupServer.

¹⁰⁵ The GroupServer Development group contains an [example of the new Invitation Response page](#).

¹⁰⁶ Sending a welcome message when joining a group will close [Ticket 303](#). Previously new members only saw a welcome message when signing up or joining a group themselves. The same fix also removed a [rather nasty hack](#).

¹⁰⁷ Logging the declined invitations closes [Ticket 278](#). It is also what allows people to be redirected if they follow invitations that have had a response. It will also add another entry in the ongoing saga “Why Physical Deletes are the Work of the Devil”.

¹⁰⁸ A simple Cron-job would probably be fine at cleaning up the unwanted profiles (see [Ticket 446](#)).

¹⁰⁹ Telling all group administrators that a new member has joined a group will close an [irritating issue with GroupServer](#).

¹¹⁰ Adding the list of invitations to the profile page will close [Ticket 347](#).

¹¹¹ Administrators being moderated only effects administrators of sites with moderated groups; regardless [Ticket 235](#) is closed.

Leave

The Leave page now presents some alternative ways of **reducing email overload**¹¹². The alternatives are going onto a **daily digest** of topics, or to participate in the group using the **web only**. We hope this will reduce the number of people who leave groups.

Join and Leave Log

People joining and leaving groups is now audited¹¹³. The audit-data is then fed into a new *Join and Leave Log*¹¹⁴.

Manage Members

The Manage Members page looks much the same as it did. Those familiar with the page will notice the **profile photos**, and a subtle rewording of the text. For example, people who have been invited to join the group are now clearly marked as such. Underneath the page has undergone a major rewrite, thanks to the heroic efforts of Alice¹¹⁵. The new page now meshes properly with invitations, so invites can be withdrawn without causing any problems¹¹⁶.

Change Site Introduction

The page that is used to change the text that appears on the homepage of the site has been updated so it uses the **WYMeditor**¹¹⁷. This is the same JavaScript-based HTML editor that is used to change a biography on the *Profile* of a member.

Share Box

The most visible change is to the topics and posts. all the *Short link* links on the topics and posts pages to a JavaScript-based **share box**¹¹⁸. The share box provides a quick and easy way to share a post or topic on Facebook, Twitter, or just as a URL.

¹¹² The alternatives to leaving are shown in an [example leave page](#) in the GroupServer Development group.

¹¹³ Auditing when people join and leave a group closes [Ticket 341](#).

¹¹⁴ The GroupServer Development group contains some examples of what the **Join and Leave Log** looks like to different people. Creating the log closes [Ticket 341](#).

¹¹⁵ Neither [Ticket 420](#), [Ticket 442](#) or the appearance of the page convey what a monumental task it was to rewrite the Manage Members page. It is now a page that can be improved, rather than a huge hack.

¹¹⁶ It would be best if I kept hacks that *used* to exist around invitations to myself. Invitations now work well, closing [Ticket 435](#).

¹¹⁷ The fix so the **WYMeditor** on the *Change the Site Introduction* page closes [Ticket 357](#).

¹¹⁸ Originally the share-box was slated for Pineapple Snow, but Richard completed the short-link improvements early on the request of a client, closing [Ticket 378](#).

Buildout

Buildout is a project designed to solve 2 problems:

1. Application-centric assembly and deployment

Assembly runs the gamut from stitching together libraries to create a running program, to production deployment configuration of applications, and associated systems and tools (e.g. run-control scripts, cron jobs, logs, service registration, etc.).

Buildout might be confused with build tools like make or ant, but it is a little higher level and might invoke systems like make or ant to get its work done.

Buildout might be confused with systems like puppet or chef, but it is more application focused. Systems like puppet or chef might use buildout to get their work done.

Buildout is also somewhat Python-centric, even though it can be used to assemble and deploy non-python applications. It has some special features for assembling Python programs. It's scripted with Python, unlike, say puppet or chef, which are scripted with Ruby.

2. Repeatable assembly of programs from Python software distributions

Buildout puts great effort toward making program assembly a highly repeatable process, whether in a very open-ended development mode, where dependency versions aren't locked down, or in a deployment environment where dependency versions are fully specified. You should be able to check buildout into a VCS and later check it out. Two checkouts built at the same time in the same environment should always give the same result, regardless of their history. Among other things, after a buildout, all dependencies should be at the most recent version consistent with any version specifications expressed in the buildout.

Buildout supports applications consisting of multiple programs, with different programs in an application free to use different versions of Python distributions. This is in contrast with a Python installation (real or virtual), where, for any given distribution, there can only be one installed.

To learn more about buildout, including how to use it, see <<http://www.buildout.org/>>.

12.1 Installation

There are a number of ways to install buildout. You can install it as you would any other package, using pip or easy_install. In this case, you'll get a buildout command that you can use to build projects. To build a project, just use:

```
buildout
```

from a project directory.

Buildout's (stubborn) philosophy, however, is that projects should be self-contained, and not require changes to a shared Python installation. To avoid changing a shared Python installation you can download a bootstrap script that, when run, will install buildout locally in your project.

The bootstrap script for buildout version 2 is at:

<https://bootstrap.pypa.io/bootstrap-buildout.py>

So, for example, to install buildout 2 in a project, you might:

```
wget https://bootstrap.pypa.io/bootstrap-buildout.py
python bootstrap-buildout.py
```

Then to build your project, you can just run:

```
bin/buildout
```

from the project directory.

buildout 2 is somewhat backward-incompatible with version 1. Most projects will probably work fine with either. If you need to keep using version 1, however, specify a version requirement when you use pip or easy_install, or use the version 1 bootstrap script at:

<http://downloads.buildout.org/1/bootstrap.py>

Resources

- Documentation: <http://groupserver.readthedocs.org/>
- Code repository: <https://github.com/groupserver/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>

A

all (None attribute), [29](#)

E

email (None attribute), [29](#)

environment variable

 PATH, [25](#)

G

groups (None attribute), [29](#)

I

id (None attribute), [29](#)

N

name (None attribute), [29](#)

O

other (None attribute), [29](#)

P

PATH, [25](#)

preferred (None attribute), [29](#)

ProfileData() (class), [29](#)

Python Enhancement Proposals

 PEP 420, [41](#)

U

unverified (None attribute), [29](#)

url (None attribute), [29](#)