Savane Installation Guide

Irene Fernández Monsalve

Savane Installation Guide

by Irene Fernández Monsalve

Revision History

Revision 1.3 18-9-2005 Revised by: IFM* *New section on external feature configuration, and links to CVS and Mailman documentation*

Revision 1.2 26-7-2005 Revised by: IFM* *New SSL config section* * *New project approval section*

Revision 1.1 24-7-2005 Revised by: IFM * *Addition of mail config section* * *Addition of Debian package notes*

Revision 1.0 15-7-2005 Revised by: IFM* *Completion of site-configuration section*

Revision 0.1 8-7-2005 Revised by: IFM * *Preliminary version*

Table of Contents

List of Examples

I. Introduction

Chapter 1. About this document

In this document, I will record my step by step local test installation and initial configuration of savane v.1.0.6, on a Debian Sarge. Wherever possible, I will give general explanations of the process, in the hope it can be useful under different situations.

The aim of the document is to provide a complete guide to savane installation and configuration, for system administrators with basic knowledge of the various services implemented, or to any user wanting to try this software.

Chapter 2. About Savane

2.1. What is Savane?

Savane is a Web-based Libre Software hosting system. Through a web-interface, and without the need of client installation, it provides an environment for collaborative work, integrating the following tools:

- Individual account maintenance
- Project and Member management
- Issue tracking (bugs, task, support)
- Mailing Lists (Mailman)
- Web page management through CVS
- Http download area
- Version Control Systems (CVS, Arch, Subversion)

2.2. Brief history

Free software has a particular engineering process, involving decentralized collaboration of many people. Special tools to allow collaborative work are needed to canalized these decentralized efforts, and provide communication channels, such as mail lists, version control and bug tracking systems.

Web-based platforms, offering these services without the need for client installation, thus originated, serving as a central point for free software development. For a long time, the SourceForge platform, powered by alexandria, has been the most important service of this kind, providing an integrated collaborative development environment.

The GNU project funded its own development plataform, Savannah (http://savannah.gnu.org), also powered by alexandria. However, shortly afterwards, VA Software Corporation decided to close alexandria code,converting it into proprietary software. Not surprisingly, this decision led to the creation of several forks with the intention of maintaining and developing the software and keeping it free.

The Savannah development project was one of them. Savannah platform maintainers decided to develop their own fork from alexandra 2.0, assuring in this way that the software would continue to evolve free, and the platform would continue to work properly. Later on, the Savannah development project changed the name of the software to savane, eliminating the ambiguity of the Savannah term (both software and hosting platform), and funding, at the same time, a new hosting site: Gna! (http://gna.org) with the support of FSF-France, where the project currently lives.

II. Savane Installation

Chapter 3. Structure of the savane application

Savane is composed of a database, a web frontend and a system backend:

3.1. The Frontend

The frontend is made up of web user interface (run by apache web server with php), and a database (mysql). All user modifications to the database are made via the web frontend, which can also trigger actions executed by the backend. The frontend+database alone will allow the creation of users and projects, preference configuration, and item (bugs, tasks..) submission and modification.

3.2. The Backend

This part of savane is a set of perl scripts and perl modules that allow the synchronization of the database with the system. These scripts will be periodically invoked by cron, and can also be called by site administrators when needed via shell.

The aim of database and system synchronization is to enable features that require system disk space or other system servers, such as download areas, or mailist creation (through mailman).

Example 3-1. Providing download area

1. Through the *web frontend* a user creates a new project, that will be registered as a new group in the database.
2. The *backend* will create a unix group corresponding to the database group, and create unix users accounts belonging to that group for each member of the project registered in the database.
3. The *backend* will then create a directory for each group at the configured location, that will serve as the assigned area for that project (group)

Chapter 4. Software requirements

4.1. General

- GNU coreutils, textutils, shellutils...
- GNU gettext
- mailx
- exim (or sendmail)
- perl >=5.6

4.2. Frontend: Web server and database

- apache
- libapache-mod-php4
- libapache-mod-ssl
- php >= 4.1.0 (with file upload activated)
- mysql-server
- php4-mysql

4.3. Backend

Perl Modules

- DBI (libdbi-perl in debian)
- DBD::mysql (libdb-mysql-perl in debian)
- Digest::MD5 (perl in debian)
- Mail::Send (libmailtools-perl in debian)

Other applications

- Mailing list manager: Mailman
- Version Control System: CVS, Arch, Subversion (one of them, or a combination)
- ViewCVS (optional)

Chapter 5. Installation

5.1. Downloading and extracting the sources

Once the necessary packages have been installed, the next step is to download the latest tarball from the CVS tree located at gna.org/projects/savane (http://gna.org/projects/savane), and unzip it in the desired directory, that will become the root of the installation.

In my test installation, the chosen directory was `/home/irene/proyecto`. After unzipping the tarball, the following directory structure and files were created inside `/home/irene/proyecto/savane/`:

```
irene@aoslox:~$ ls /home/irene/proyecto/savane
AUTHORS              COPYING.documentation  INSTALL.verbose
AUTHORS.Savannah     COPYING.software       lib
AUTHORS.SourceForge  CVS                    Makefile
backend              db                     Makefile.in
ChangeLog            debian                 po
ChangeLog.bak        doc                    README
ChangeLog.byversion  etc                    REQUIREMENTS
configure            frontend               update
configure.cache      INSTALL                www
```

In addition to usual package files and directories, we can see some particular directories:

- *backend*

  Contains backend perl scripts (all savane scripts' names start with sv_)

- *lib*

  Contains the savane lib `Savannah.pm`

- *frontend*

  Contains php files, as well as some perl scripts needed for the frontend

- *www*

  symlink to `frontend/php/`, where the main user interface `index.php` lives.

- *etc*

  Contains default crontab and logrotate files, needed for the backend, as well as a `site-specific-content` directory which allows php and html modifications of site content

5.2. Installing the package

The following should be done:

```
./configure
#build what need to be built
make
```

```
#create database
make database
#create configutation file
make conf
#install backend scripts and translation files
make install
```

We will now examine more closely each of these steps.

5.2.1. `./configure`

Firstly, **`./configure`** checks necessary packages are installed:

```
irene@aoslox:~/proyecto/savane$ ./configure
checking for which... yes (non-gnu)
checking for make... yes
checking for bash... yes
checking for perl... yes
checking for sed... yes
checking for find... yes
checking for gettext... yes
checking for mysql... yes
checking for mailman... yes
checking for automake... yes
checking for autoconf... yes
checking for makeinfo... yes
checking for gzip... yes
checking for GNU/Linux distribution... debian-based
```

We are then asked if we would like the process to be interactive, thus setting certain configuration values:

1. URL of savane directory

   ```
   What is the URL directory of Savane on your http server?
   ex: for sv.gnu.org/ the answer is /
   ex: for bla.org/savane/ the answer is /savane
   [/]: /
   ```

   This will depend on how apache is configured. If the savane web interface is configured as a virtual host, the answer will be `/`. On the other hand, if it is configured as a subdirectory of the home URL, the answer will be `/directoryname`.

2. Paths to configuration files, binaries, perl modules, and i18n files. Leaving blank answers will load the defaults:

   ```
   Where should we store configuration files?
   Choose one of the following.
   ex: /etc/savannah
   (WARNING: if you use something else than the default setting, be sure to set
   SAVANE_CONF environment variable in both Apache and terminals you use, it is
   required for the frontend and the backend)
   [/etc/savannah]:
   ```

```
Where should we install binaries and scripts?
ex: /usr/bin
ex: /usr/local/bin
[/usr/local/bin]:

Where should we install perl modules?
ex: /usr/local/lib/site_perl
[/usr/local/lib/site_perl]:

Where should we install i18n files?
ex: /usr/share/locale
ex: /usr/local/share/locale
[/usr/share/locale]:
```

3. We are finally asked about database name, and any other options for mysql:

```
What is the database name?
ex: savane
[savane]:

Any options to use for ?
ex: you may add for mysql, if you do not have a ~/.my.cnf
something like -u user -p
(never include password in clear text here)
options:
```

This process finishes with the creation of `Makefiles`:

```
creating ./backend/Makefile...
creating ./Makefile...
creating ./db/Makefile...
creating ./doc/Makefile...
creating ./doc/devel/Makefile...
creating ./etc/Makefile...
creating ./frontend/php/css/Makefile...
creating ./frontend/php/docs/Makefile...
creating ./frontend/php/images/Makefile...
creating ./lib/Makefile...
creating ./po/Makefile...
```

5.2.2. `make`

`make` and **`make database`** will build several minor parts (translation files, etc), and create the database. **`make conf`** will generate the configuration files, that will be studied in the next chapter.

5.2.3. `make install`

`Make install` will install backend scriptpts. It must be run even if you only want to use the frontend, since some of the scripts, like sv_update_conf or sv_skill are used for development purposes and must be installed.

The scripts installed in this step can be found at `savane/backend/`, distributed amongst the following directories:

- `accounts`

  Scripts related to accounts management. These scripts can be used to create unix groups and users on a system with the info in the database. It can also build repositories of different kind.

- `admin`

  Scripts designed for savane administrators (of an installation). For instance, sv_register_discard permit to remove a group from the database completely.

- `cvs`

  Scripts for CVS management.

- `darwin`

  Scripts for savane use on Darwin/Mac OS X.

- `devel`

  Scripts for savane developers.

- `install`

  Scripts for savane installation/upgrade.

- `mail`

  Scripts related to mail management.

Translation files will be installed at this step as well. Note that transaltion will only work if your server is configured to accept different locales. On Debian, run **`dpkg-reconfigure locales`** to add more locales.

Once this step is accomplished, you can run sv_skill, that will only work if the previous step has been completed, and thus serve as a check of successful installation. This script will load form the CJN (http://sf.net/projects.cjn) project a list of "skills", that will be presented to each platform user on their profile page, so they can edit a resume about themselves.


In order for this script to work, you must have an active internet connection, so the information can be downloaded.

An **`anoncvs`** user and group must be created to allow the backend cvs functions to operate.

To wrap up the backend installation, take a look at `/etc/cron.d/savane` and adapt it to suit your needs, since this single file runs all the backend jobs.


For Debian users:

The Savane project also offers debian packaged versions of the software. To use the stable repository, add in your `/etc/apt/sources.list` the following line:

```
deb http://dl.gna.org/savane/debian stable/
```

After running **`apt-get update`**, several savane packages will be available, corresponding to the frontend and different backend parts. The "savane" package will install all the others, and if no config exists, will run the Savane interactive configuration tool (Described in the next section).

Chapter 6. Savane configuration

Savane relies on two configuration files, `savannah.conf.pl` and `savannah.conf.php`. The second is automatically generated from the first, and must not be edited by hand. The default is to generate them at `/etc/savannah`, and if a different location is chosen it will be necessary to set the `SAVANE_CONF` environment variable to the path, both in Apache and in terminals used to invoke backend scripts.

During our installation, after typing **`make conf`**, both files will be generated with the options obtained through an interactive process with the following questions:

1. Host names and port installation:

```
***** /etc/savannah/savannah.conf.pl (re)creation ******

1) Default hostname
(It must be the naked form of the domain)
Ex: savannah.gnu.org
[aoslox]: proyecto.savane.net

2) HTTPS hostname
(It must be the naked form of the domain.
If you do not have https server, comment out)
Ex: savannah.gnu.org, $sys_default_domain
[OUT]: proyecto.savane.net

3) HTTPS port
(Port number. If you do not have https server, comment out)
Ex: 443
[OUT]: 443

4) Brother hostname
(You can run Savannah with two different domain names. You'll be able to write a different co
nfiguration for each one. The two brother/companion sites will share the same database. /etc/
savannah/local.inc.pl is far more important since the default access to the database is done
using this one. Here you can let your savannah installation aware of the existence of a broth
er/companion site, so while people login, it will be allowed to them tologin on both site in
one click. If you do not have brother/companion site, comment out. If you do not understand w
hat is it about, you probably do not need that feature, comment out.)
Ex: savannah.nongnu.org
[OUT]:
```

   In brackets, we are given default values. `OUT` will comment that option, thus disenabling it. In this installation, I enabled https, with the same host name as the default host. This configuration allows savane to work both through http and https: login will be done through https, thus securing passwords, and the user will be able to chose whether to stay in SSL afterwards or not. The brother hostname feature is inherited from the Savannah platform, that offers hosting to GNU and non-GNU libre software projects, separating both sites.

2. Database:

```
6) SQL database name
(--)
Ex: savane
```

```
[OUT]: savane

7) SQL database user
(--)
Ex: mysqluser
[OUT]: mysqluser

8) SQL database password
(--)
Ex: mysqlpasswd
[OUT]: mysqlpasswd

9) Additional database settings
(param=value pairs separated by colons, eg. param1=value1:param2=value2...)
Ex: mysql_socket=/non-standard-path-to/mysqld.sock
[OUT]:
```

3. Installation paths:

```
10) Local dir of the installation of Savannah, in which is the PHP frontend,
 in subdirectory ./frontend/php
(IT MUST BE AN ABSOLUTE PATH NAME)
Ex: /usr/local/sv/src/savannah
[/home/irene/proyecto/savane]:

11) Default web directory
(Suffix appended to the default domain)
Ex: /
[/savane]:/


12) Local dir of the site-specific content
(--)
Ex: /etc/savannah/savannah-content
[$sys_topdir/etc/site-specific-content]:


13) Misc directory
(Local dir where dumps will be stored etc)
Ex: /subversions/sv
[OUT]: /subversions/sv
```

The default web directory will depend on whether we configure apache to host the platform as a virtual host, with its own domain (**/**) or as a subdirectory of the home URL (**/dirname**). See the apache configuration section for more details.

4. Platform customization:

```
14) Platform name
(Name shown on public pages for the whole service)
Ex: A New Savane Installation
[OUT]: Instalación de Savane y Documentación

15) Server administration project unix name
(Unix group name of the meta-project used for administration. Take care to avoid
```

```
conflicts with group name existing on your system, take care to select a valid
unix group name: no checks will be done for this project unixgroup name.)
Ex: admin
[OUT]: admin

16) Default locale
(It must be a valid locale name)
Ex: french
[OUT]: es_ES

17) Date format
(Date formatting. If you want to use the default locale formating, comment out
(it's usually a good idea))
Ex: Y-M-d H:i
[OUT]:

18) Default theme
(--)
Ex: savannah
[OUT]: savannah

19) Logo name
(The engine will search for a file like
savannah/frontend/php/images/$theme.theme/$sys_logo_name.
If you do not want any logo, comment out)
Ex: floating.png
[OUT]:
```

In this section, question 15 is very important. In order to do fine configuration and administer the platform, it will host a special kind of project: a site administration project, whose unix_name must be defined before it is created in the configuration file. This way, once the first project is created, being given this same unix_name, it will have special administration sections, and the user who created will be able to become superuser and administer the server from the admin server space.

Regarding themes, different `.css` files can be found at `frontend/php/css`, that can generate different styles.

5. Mail settings

```
20) Mail domain
(--)
Ex: gnu.org, $sys_default_domain
[OUT]: localhost

21) Admin mail address
(The mail domain we'll be added to this username)
Ex: savannah-hackers
[OUT]: admin

22) No reply address used in the trackers
(--)
Ex: NO-REPLY.INVALID-ADDRESS
[OUT]: NO-REPLY.INVALID-ADDRESS
```

```
23) [BACKEND SPECIFIC] List of emails
(If you do not want such file to be updated by the backend, comment out)
Ex: /etc/email-addresses
[OUT]: /etc/email-addresses

24) [BACKEND SPECIFIC] List of emails aliases
(If you do not want such file to be updated by the backend, comment out)
Ex: /etc/aliases
[OUT]: /etc/aliases
```

6. PAM and Kerberos support

```
25) PAM support
(AFS, Kerberos (...) authentication can be made via PAM)
Ex: no
[OUT]: no

26) Kerberos 5
(If you do not know what it is about, you surely don't have to deal with a kerberos
server, say no here.)
Ex: no
[OUT]: no
```

7. Backend configuration for user accounts

```
27) [BACKEND SPECIFIC] User home directory
(Usually /home. You can uncomment if you do not plan to provide accounts)
Ex: /home
[OUT]:/home/savane/users

28) [BACKEND SPECIFIC] User home directory subdirs
(Users home is by default /home/user. If you set this to 1, you'll get /home/u/user,
and if you set it to 2, you'll get /home/u/us/user.
It may be very convenient if you have plenty of users.)
Ex: 2
[OUT]:1

29) [BACKEND SPECIFIC] User default shell
(cvssh is a limited shell, choose /bin/bash if you want to provide full access to your users)
Ex: /usr/local/bin/sv_membersh
[OUT]: /usr/local/bin/sv_membersh

30) [BACKEND SPECIFIC] Prefix for user* binaries
(If you do not want to use useradd/usermod/userdel that are in the usual PATH
but specific ones, you can type here their prefix. Otherwise, comment out)
Ex: /usr/local/savannah/bin
[OUT]:
```

In this section, we can configure the shell users will have access to, the ubication of home directories, and the possibility of creating them in alphabetically ordered subdirectories

8. Backend system syncronization with database through cron

```
31) [BACKEND SPECIFIC] Cron job: related to users
(If you do not want your system to be synchronized with database automatically
regarding to users infos (/home/, /etc/passwd)), comment out)
```

```
Ex: yes
[OUT]: yes

32) [BACKEND SPECIFIC] Cron job: related to groups/projects
(If you do not want your system to be synchronized with database automatically
regarding to groups infos (/etc/group), comment out)
Ex: yes
[OUT]: yes

33) [BACKEND SPECIFIC] Cron job: related to viewcvs ignore list
(If you do not want your system to be synchronized with database automatically
regarding to viewcvs forbidden list, comment out)
Ex: yes
[OUT]: yes

34) [BACKEND SPECIFIC] Cron job: related to mails
(If you do not want your system to be synchronized with database automatically
regarding to mail infos (/etc/aliases...), comment out)
Ex: yes
[OUT]: yes

5) [BACKEND SPECIFIC] Cron job: related to mailman
(If you do not want your system to be synchronized with database automatically
regarding to mailman list (it assume you have mailman installed on this system), comment out)
Ex: yes
[OUT]: yes
```

Through these options, the extent to which cron will synchronize database and system can be regulated, thus controlling the backend jobs.

9. Other backend cron jobs:

```
36) [BACKEND SPECIFIC] Cron job: database cleaning
(A special backend script will clean regularly the database. If you do not want
that cleaning to be done, comment out.
It is recommended to use it, even if your installation use no other backend tool)
Ex: yes
[OUT]: yes

37) [BACKEND SPECIFIC] Cron job: trackers reminder
(A special backend script will check regularly the database and send email to
users in defined cases. An user can decide to receive regularly task assigned
to him in a batch ; a project administrator can decide to make people that got
item with high priority not closed receiving a batch. Also, when an item is
supposed to start of to finish, a reminder should be sent to anybody supposed to get
notification for the item)
Ex: yes
[OUT]: yes
```

10. Final backend options:

```
38) [BACKEND SPECIFIC] Viewcvs configuration file
(Path to the viewcvs conffile. If you do not use viewcvs or if you do not want the
fordibben setting of this configuration file to be edit by Savannah, comment out)
Ex: /etc/viewcvs/viewcvs.conf
```

```
[OUT]: yes

39) Google search
(Add a search via google option to the search module. If you do not want
this search facility, comment out)
Ex:
[OUT]:

40) Local Administration Documentation File
(Will make available the content of a specific file to site admins.
Give the path to the file or comment out)
Ex:
[OUT]:
```

The `savannah.conf.pl` can be edited by hand at any time, typing **`make update-conf`** afterwards so that the `savannah.conf.php` is re-generated, through the script sv_update_conf (the script can also be directly invoked via a shell).

Chapter 7. Additional configurations

7.1. Apache web server

For the apache-php frontend to work properly, we must:

1. Configure PHP to accept file upload, and activate **`register_globals`**, editing the PHP configuration file, `php.ini`:

   ```
   irene@aoslox:~$ less /etc/php4/apache/php.ini
   (...)
   ;;;;;;;;;;;;;;;;
   ; Data Handling ;
   ;;;;;;;;;;;;;;;;
   (...)
   register_globals = on
   (...)
   ;;;;;;;;;;;;;;;;
   ; File Uploads ;
   ;;;;;;;;;;;;;;;;

   ; Whether to allow HTTP file uploads.
   file_uploads = On
   ```

2. Configure apache (editing `httpd.conf`) to handle and work with php files:

   - Make sure the php module is loaded:

     ```
     irene@aoslox:~$ less /etc/apache/httpd.conf
     LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
     ```

   - Let apache search for `index.php` as well as `index.html`:

     ```
     irene@aoslox:~$ less /etc/apache/httpd.conf
     DirectoryIndex index.html index.php
     ```

   - Files with `.php` extension must be associated to a `x-httpd-php application`:

     ```
     AddType application/x-httpd-php .php
     AddType application/x-httpd-php-source .phps
     ```

Now we can add the necessary lines to allow apache to load savane's web interface. Whether we are configuring the interface as a Virtual Host, or as a subdirectory of our home URL, two steps must be taken:

1. Set SAVANE_CONF environment variable.

   Make sure the `env_module` is being used by apache:

   ```
   LoadModule env_module /usr/lib/apache/1.3/mod_env.so
   ```

This module provides for modifying the environment which is passed to CGI scripts and SSI pages, and will allow us to set the SAVANE_CONF environment variable to the path of our savane configuration files:

```
<IfModule mod_env.c>
SetEnv SAVANE_CONF /etc/savannah
</IfModule>
```

Note that this directive must be set either in the Virtual Hosts section, or in the main server section.

2. Activate `.htaccess`, with `AllowOverride` directive.

   Inside the php frontend directory structure, there are two `.htaccess` files, containing `ForceType` directives (`savane/frontend/php/project/stats`, and `savane/frontend/php`), forcing certain files to be treated as x-httpd-php applications, regardless of the extension.

   `AllowOverride` must be placed inside a `Directory` directive, in the `main server` or `Virtual Host` section:

   ```
   <Directory /home/irene/proyecto/savane/frontend/php>
   AllowOverride All
   </Directory>
   ```

   If you do not want to activate `.htaccess`, you should add something like the following, where PATH is the subdirectory where savane is installed (empty if it is in the top directory):

   ```
   <Location PATH/users>
   SetHandler application/x-httpd-php
   </Location>
   ```

Putting all this together, in a virtual host installation of the frontend, the section would end up looking something like this:

```
NameVirtualHost 192.168.62.11
<VirtualHost 192.168.62.11>
ServerName savane.proyecto.net
DocumentRoot /home/irene/proyecto/savane/frontend/php
ServerAdmin your.name@my.net

<Directory /home/irene/proyecto/savane/frontend/php>
AllowOverride All
</Directory>

<IfModule mod_env.c>
SetEnv SAVANE_CONF /etc/savannah
</IfModule>
</VirtualHost>
```

On the other hand, if the frontend is installed as a subdirectory of your home URL, for instance, http://myhome.net/savane/, it will be necessary to add an `Alias` pointing to the web interface directory, in addition to the `AllowOverride` and `SetEnv` directives:

```
Alias /savane /home/irene/proyecto/savane/frontend/php
```

7.2. Database

After configuring the server, the main frontend page informed me of a problem with the database, to which it had been unable to connect. I therefore tried to access the database with the user and password I had configured during the installation, via a shell, but found that user wasn't allowed to access the savane as database. To fix this, I entered the database as root and granted `mysqluser` privileges to access `savane` database:

```
irene@aoslox:~$ mysql -h localhost -u root
mysql>GRANT ALL on savane.* TO mysqluser@localhost;
```

And then entered the database as **`mysqluser`** and set the password to **`mysqlpasswd`**:

```
irene@aoslox:~$ mysql -h localhost -u mysqluser -p savane
mysql>SET PASSWORD FOR mysqluser@localhost=PASSWORD('mysqlpasswd');
```

If you installed the Debian package, you will also need to configure apache and mail as described in this chapter, and, in addition, create the database. Although the package contains the necessary files, it does not build it.

To manually create the database, you must:

```
# uncompress sql files
gunzip /usr/share/doc/savane-database/1.database_savane.structure.sql.gz
gunzip /usr/share/doc/savane-database/2.database_savane.initvalues.sql.gz

# create database
mysql -e "CREATE DATABASE databasename"

# create database structure
mysql databasename < /usr/share/doc/savane-database/1.database_savane.structure.sql
mysql databasename < /usr/share/doc/savane-database/2.database_savane.initvalues.sql
```

7.3. Mail transfer agent

In order to function correctly, savane needs a mail transfer agent installed and configured. Mails are a necessary part of user registration process, being sent to user's address with a confirmation URL, as well as being used for several other notifications.

In my test installation, I used exim4. Running **`dpkg-reconfigure exim4-config`** will guide you through the configuration steps. I set it to initiate SMTP connections to remote sites directly (**`Internet site`** option), assigned **`savane.proyecto.net`** as the mail name, and set IP addresses to listen on to **`127.0.0.1`**. I left the default values for the rest of the options.

You can test exim by sending an email to an address you can check:

```
aoslox:/# exim4 -v your.mailbox@dom.ain
From: irene@savane.proyecto.net
```

```
To: your.mailbox@dom.ain
Subject: Testing exim
Testing exim
.
```

7.4. Secure server with mod-ssl

To set up a secure webserver (https) for savane, I used apache and mod-ssl (`libapache-mod-ssl` package in Debian).


Make sure your savane configuration enables https. Your `savannah.conf.pl` should include a section like this:

```
# Default HTTPS domain
      # It must be the naked form of the domain
      # Ex: "savannah.gnu.org", "443"
       our $sys_https_host="savane.proyecto.net";
       our $sys_https_port="443";
```

To get the secure web server working after installing mod-ssl you must generate an SSL certificate, and configure apache to use it.

7.4.1. SSL certificate creation

In my test installation, I generated a test certificate, through the **`mod-ssl-makecert`** command, that guides you through an interactive process:

```
What type of certificate do you want to create?
  1. dummy     (dummy self-signed Snake Oil cert)
  2. test      (test cert signed by Snake Oil CA)
  3. custom    (custom cert signed by own CA)
  4. existing  (existing cert)

Use  dummy     when you are a vendor package maintainer,
     test      when you are an admin but want to do tests only,
     custom    when you are an admin willing to run a real server
     existing  when you are an admin who upgrades a server.

Normally you would choose 2.
your choice: 2

Which algorithm should be used to generate required key(s)?
  1. RSA
  2. DSA

Normally you would choose 1.
```

```
your choice: 1

SSL Certificate Generation Utility (mkcert.sh)
Copyright (c) 1998-2000 Ralf S. Engelschall, All Rights Reserved.

Generating test certificate signed by Snake Oil CA [TEST]
WARNING: Do not use this for real-life/production systems
```

I chose to create a "test" certificate, without validity for real-life systems, but suitable for test purposes.

The certificate generation process will then enquire about general information to be included in the certificate, such as the domain it will serve (savane.proyecto.net in my case); and whether you want to encrypt your private key with a passphrase, to which I answered ’yes’.

7.4.2. Configuring apache

First, make sure Apache uses the mod-ssl module:

```
LoadModule ssl_module /usr/lib/apache/1.3/mod_ssl.so
```

Next, configure Apache to listen upon the SSL port (443), and add SSL options for your configuration. I included the following lines in `httpd.conf` just before my first virtual host section:

```
<IfModule mod_ssl.c>

##
##  SSL Global Context
##
# I want apache to listen to 443 port in addition to the standard 80.
# This will allow me to later configure virtual hosts for
# savane.proyecto.net listening to each port, and answering petitions
# for http://savane.proyecto.net as well as https://savane.proyecto.net

Listen 80
Listen 443

#
#   Some MIME-types for downloading Certificates and CRLs
#
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#   Pass Phrase Dialog:
SSLPassPhraseDialog  builtin

#   Inter-Process Session Cache:
SSLSessionCache         dbm:/var/run/ssl_scache
SSLSessionCacheTimeout  300

#   Semaphore:
SSLMutex  file:/var/run/ssl_mutex
```

```
#   Pseudo Random Number Generator (PRNG):
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>
```

Finally, add a new Virtual Host section, to make the savane platform available through https. To the savane specific configurations, SSL directives are also added:

```
NameVirtualHost 127.0.0.1:443
<VirtualHost 127.0.0.1:443>

  ServerName savane.proyecto.net
  DocumentRoot /home/irene/proyecto/savane/frontend/php
  ServerAdmin irene@haicku

  <Directory /home/irene/proyecto/savane/frontend/php>
    AllowOverride All
  </Directory>

  <IfModule mod_env.c>
    SetEnv SAVANE_CONF /etc/savannah
  </IfModule>

  <IfModule mod_ssl.c>
    SSLEngine on
    SSLCertificateFile    /etc/apache/ssl.crt/server.crt
    SSLCertificateKeyFile /etc/apache/ssl.key/server.key
    SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
  </IfModule>

</VirtualHost>
```

 Keeping two virtual hosts sections for the platform (one at port 80, another at port 443) allows platform users to chose wether to stay in SSL mode after logging, or not.

7.5. External features: Mailman and CVS

Savane relies on Mailman and CVS to provide mail list and source code management. As a result, they must be installed and configured in your system independently. Explaining this process escapes the scope of this document, but will probably entail getting them packaged for your distribution, and using the tool provided to configure them. In the case of Debian, this would mean running **`dpkg-reconfigure cvs`** and adding 2 new directories, one for Source Code, and another for HTML (homepage) management.

For further information, you can take a look at:

- Exim and Mailman HOWTO (http://www.exim.org/howto/mailman.html)

- CVS Manual (http://ximbiot.com/cvs/manual/)

III. Site Configuration

Chapter 8. Bootstrapping the Site

Up to this point, savane's frontend and backend have been correctly installed and configured. However, in order for the site to start offering services, it must also be configured, and maintained.

Administration of the site takes place through a special kind of project hosted by the own site. This project must have the unix_name given during the configuration, and must be the first to be created.

Next steps, therefore, will consist in the creation of the first user, that will become the site administrator, and the first project, that will become the site administration project.

User registration is a straightforward process, that only requires the introduction of user name, password, e-mail address, and real name. The backend then triggers a confirmation mail to the given address, with an URL that must be visited to activate the registration.

The registered user can then create the first project, that will become the site administratioin interface. New project creation involves five steps:

1. Hosting services and requirements
2. project description and dependencies
3. project full and unix names (unix_name will be **`admin`** in my case)
4. choice of license
5. project type (only `default`) will be available this time

Unlike ordinary projects, this one doesn't need approval, and will be automatically generated. The user that created it now becomes the site administrator.

At this point, the site is operative, although further configuration must be done through the administration project in order for all the project services to be tuned to your needs, and be fully operational. We will see this process in the following chapters.

Chapter 9. How the site works

Let’s continue with a little bit of theory about how it all works, how site is configured, and how users and groups work with frontend and backend.

9.1. Site administration

Site administration takes place at two levels. On one hand, we have already seen how the first registered user becomes *site administrator*, through the first project created, that becomes the *site administration project*. The administration team can then become members of the group, having access to all the tools savane offers for collaborative work. In addition, the administration project pages will provide an interface for communication between platform users and administrators, through the public project areas, and tools such as the support tracker.

On the other hand, we have *server administration*. Server administration can only be done by a superuser (site administrator can become superuser at any time), and involves, amongst others, project approval and lower-level server configuration. After following all the steps previously outlined, the site is still not fully operational: server configuration must be done in order for all the subsystems to work. We will see how this, and project approval, is done in next chapters.

9.2. User and group interaction with the backend

It is important to distinguish between:

- *Registered users NOT members of any group*

  When a user is registered at the savane-powered site, the corresponding data will be stored in the database (frontend), but no backend action will be triggered.

  As a result, that user will have potential access to frontend services (bug/task/support submission) and read- only acess to public areas, but will not be able to use backend functionalities.

- *Groups=Projects*

  When a new project is registered at the site, the backend comes into action. Firstly, the script `sv_groups` creates a group under the unix_name of the project in the system running savane. Secondly, and according to the group_type it belongs to, certain backend services will be activated, such as CVS repository or download area creation.

- *Registered users members of a group*

  It is only when a registered user becomes part of a group (either creates it, or is incorporated into an existing one) that access to the backend, and therefore to the system, is granted.

  The `sv_users` script will trigger the creation of a new user account in the system, having a home directory at the location specified in `Savannah.conf.pl`, and belonging to the the system group of the project it is part of.

  User’s home directories are created with the following structure:

```
irene@aoslox:~$ ls -la /home/savane/users/i/irenefm/
drwxr-sr-x  4 irenefm svusers 4096 2005-07-11 12:44 .
drwxr-sr-x  3 root    staff   4096 2005-07-11 12:44 ..
drwxr-xr-x  2 irenefm svusers 4096 2005-07-11 12:44 .gnupg
-rw-r--r--  1 irenefm svusers    0 2005-07-11 12:44 .savane
drwxr-xr-x  2 irenefm svusers 4096 2005-07-11 12:44 .ssh
```

- *Group Type*

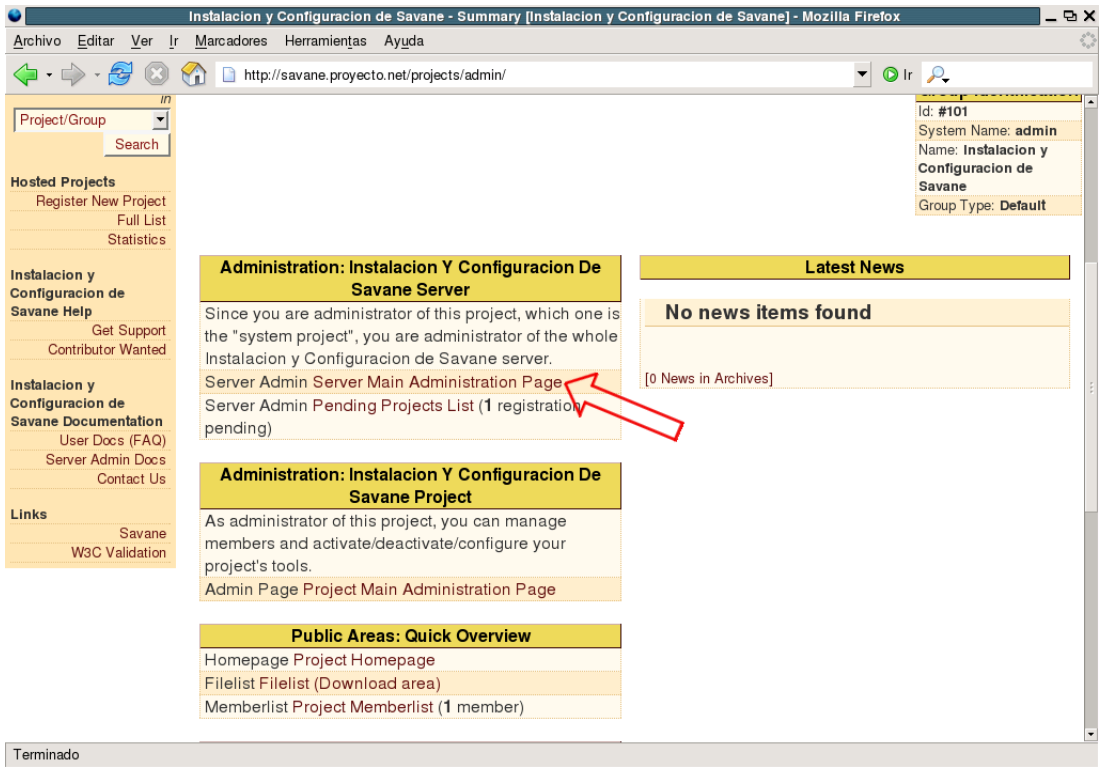
  Every project (group) must fall into a *group type* category. Initially only "default" is available, and it is the sever administrator's task to define the different group types the site will need, and the different subsystems each will offer. We will see this process in the next chapter.

Registered users members of a group will be able to use their system account to access backend services via shell. Access is granted through SSH, but password authentication is disabled. The only accepted method is ssh-keys. As a result, an ssh-key pair must be generated by the user, and the public key must be registered through the site, in the Account Conf section. This will trigger further backend action, copying the key into the user's `.ssh` directory, in a file named `authorized_keys`:

```
irene@aoslox:~$ ls /home/savane/users/i/irenefm/.ssh/
authorized_keys
```

Chapter 10. Group type configuration

Logging in as the site administrator, we will see a menu that as well as allowing us to "logout", or visit "my items", offers the possibility to become superuser. This is necessary for the rest of steps we'll take, at the server administration level. Opening the site administration project, ("Instalación y configuración de savane" in my case), we can see the option to enter the server administration area:

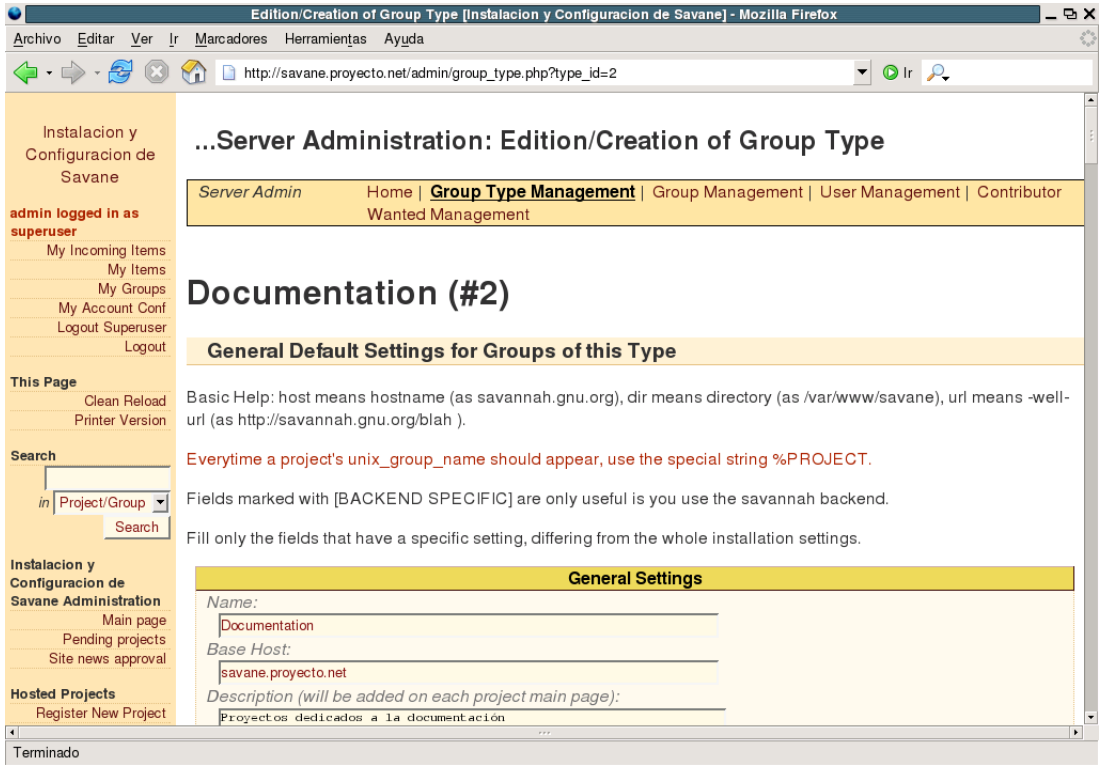


Once in the server administration area, we are informed that:

> It is highly recommended to make sure Group Type Configuration is correct, especially on a fresh install or after upgrade.

All projects must fall into a "Group type" category. On a fresh install, only "default" type is available, but it is not configured. The server's administrator's first task will be to design the different group types the site will offer, the different services each will provide, and the disk space each will use.

Even if only one group type is needed, the values for the "ready-made" default type must be adjusted, through the Server Admin->Group type management menus. Clicking on #1 Default, we will access the settings edition page:



Among group type configuration sections, we will find two kinds of services: those offered by frontend alone, and those involving backend and system disk space or other system servers. The first kind have a trivial configuration, offering activiating/desactivating options, and different access privilege configuration (non-registered users, registered users, members of the group). On the other hand, the second kind will have a little bit more configuring to do, mainly:

- Location in the system's directory structure

  For repository, file download, and home page creation, disk space must be assigned. The location must be specified, and the type of directory-making function that will be used (Basic directory, CVS directory, Subversions directory...) must be chosen.

- Web access to the subsystem

  Whether the service administration can be done through a web interface, without the need of ssh access (Mailman), or whether web interfaces are available to browse content (Viewcvs), we will be asked the URL to access these web services.

We will now walk through the different sections, creating a documentation group type.

10.1. General Options

10.1.1. General settings

We are asked the following information:

- Name:

  `documentation`

- Base Host:

  `savane.proyecto.net`. In the case of having configured brother hosts, we would now chose which one "documentation" type would belong to.

- Description (will be added on each project main page):

  `documentation projects`

- Admin Email Address:

10.1.2. Licences

This is useful if you want project to select a license on submission. Edit site-specific-content/hashes.txt to define the list of accepted licenses.

Options are only activate/deactivate

10.1.3. Development status

Offer development status categories for projects to show on their sites.

Options are only activate/deactivate

10.2. Backend sub-systems

In this section, extracts from `httpd.conf` are included, since apache configuration is needed for the web service access. These configurations can always be done in two ways: as part of main server structure (creating aliases, and with URLs such as `http://myweb.mine/service`), or in virtual hosts sections, with URLs such as `http://service.myweb.mine`.

In the filesystem path options, we will often want the directory to be created by the backend to be named after the project: we can do this using %PROJECT special string. The path leading to that directory, however, must already exist.


In my local installation, I didn't have a DNS server configured, so for each virtual host created I had to introduce a line in `/etc/hosts`:

```
127.0.0.1       aoslox.haicku.net
127.0.0.1       savane.proyecto.net
127.0.0.1       downloads.savane.proyecto.net
127.0.0.1       cvs.savane.proyecto.net
127.0.0.1       home.savane.proyecto.net
```


If you want to see the efect these configurations have on your system, you can either wait for the next cron job, or execute **`sv_groups`**.

10.2.1. Project WWW Home Page

Savane provides projects with the option of hosting their own web pages. This section is easily configured. We are asked where in the filesystem the project homepages will live, what URL will access them, and which SCM will be used to fill it with content. The SCM selection will only affect the content shown by the frontend related to the homepage management, and CVS, Arch, or Subversion are the options offered.

I configured a Virtual Host section in apache's `httpd.conf`:

```
NameVirtualHost 127.0.0.1
<VirtualHost 127.0.0.1>
ServerName home.savane.proyecto.net
DocumentRoot /home/savane/homepages
</VirtualHost>
```

And configured the section as follows:

- Selected SCM:

  `CVS`
- Directory Creation

  `Basic directory`
- Homepage Dir (path on the filesystem) [BACKEND SPECIFIC]:

  `/home/savane/homepages/%PROJECT`
- Homepage URL:

  `http://home.savane.proyecto.net/%PROJECT`
- Homepage CVS view url (webcvs, viewcvs):

  I did not activate this option. Viewcvs configuration is explained later in Source Code Manager Section.

10.2.2. Source Code Manager (primary-CVS, secondary-Arch, tertiary-Subversion)

Savane backend will create a repository (CVS, Arch or Subversion) for each project at the specified location. The repository can be accessed via shell, and in the CVS sections of the project pages we can find indications about the specific commands to use.

In my test installation, I configured a CVS repository with Viewcvs to provide http repository browsing. Viewcvs can work through its own standalone daemon, viewcvsd listening at port 7467, or via apache and CGI. In order to use the second option, the package viewcvs-query must also be installed.

I chose this second option, configuring a virtual hosts section in `httpd.conf` such as this:

```
<VirtualHost 127.0.0.1>
ServerName cvs.savane.proyecto.net
 DocumentRoot /usr/lib/cgi-bin/viewcvs.cgi
 Alias /doc/viewcvs/ /usr/share/doc/viewcvs/html/
ServerAdmin irene@haicku
</VirtualHost>
```

I also had to do additional configuration on `/etc/viewcvs/viewcvs.conf`, activating `root_parents`:

```
# The 'root_parents' setting specifies a list of directories in which
# any number of repositories may reside.  Rather than force you to add
# a new entry to 'cvs_roots' or 'svn_roots' each time you create a new
# repository, ViewCVS rewards you for organising all your repositories
# under a few parent directories by allowing you to simply specifiy
# just those parent directories.  ViewCVS will then notice each
# repository in that directory as a new root whose name is the
# subdirectory of the parent path in which that repository lives.

root_parents= /home/savane/cvs-repositories : cvs
```

My options for this section were, therefore:

- Directory Creation

  `CVS directory`

- Repository Dir (path on the filesystem) [BACKEND SPECIFIC]:

  `/home/savane/cvs-repositories/%PROJECT`

- Repository view URL (cvsweb, viewcvs, archzoom...):

  `http://cvs.savane.proyecto.net/?root=%PROJECT`

10.2.3. Download area

Provides download areas created by the backend. In my test installation, I decided each project would have its own directory, inside a `downloads` directory in the own frontend's `files` directory.

In order to access these repositories directly via web as well, I configured a virtual host section in apache's `httpd.conf`, allowing the server to return a formatted list of directories when no index is found:

```
<VirtualHost 127.0.0.1>
ServerName downloads.savane.proyecto.net
DocumentRoot /home/irene/proyecto/savane/frontend/php/files
<Directory /home/irene/proyecto/savane/frontend/php/files>
Options Indexes
</Directory>
</VirtualHost>
```

My options, therefore, were:

- Directory Creation

  `Basic directory`

- Repository Dir (path on the filesystem) [BACKEND SPECIFIC]:

  `/home/irene/proyecto/savane/frontend/php/files/downloads/%PROJECT`

- Repository URL:

  `http://downloads.savane.proyecto.net/%PROJECT`

10.2.4. Mailing lists

The mail list server used by savane is mailman. Mailman offers a web interface for configuration, so in our savane server we will have to specify the URLs to mailman's administration and information interfaces.

In my directory tree, they are located at `/usr/lib/cgi-bin/mailman`:

```
irene@aoslox:~$ ls /usr/lib/cgi-bin/mailman/
admin    confirm  edithtml  options  rmlist  savannah
admindb  create   listinfo  private  roster  subscribe
```

And the mailing list archives at `/var/lib/mailman/archives/public`.

Thanks to a `cgi-bin` ScriptAlias, and a `pipermail` alias in apache's `httpd.conf`, both the archives and administration interfaces can be accessed via the web server, through URLs like `mainurl.mine/cgi-bin/mailman` and `mainurl.mine/pipermail/`.

Extracts from `httpd.conf`:

```
<IfModule mod_alias.c>
     Alias /pipermail/ /var/lib/mailman/archives/public/
(...)
<IfModule mod_alias.c>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
```

In configuring this section, we are also asked to set patterns for the mailists' names and addresses, being able to use special strings like %LIST, or %PROJECT, that will be replaced by their value. %NAME represents the part of the name the project admin will be asked for when creating the list. This is how I configured this section:

- Mailing-list address (would be %LIST@gnu.org for GNU projects at sv.gnu.org):

  `%LIST@savane.proyecto.net`

- Mailing list name format:

  `%PROJECT-%NAME`

- Listinfo Url:

  `http://aoslox.haicku.net/cgi-bin/mailman/listinfo/%LIST`

- Subscribe Url (for majordomo at CERN, it is majordomo_interface.php?func=subscribe&list=%LIST&mailserver=listbox.server@cern.ch):

  `http://aoslox.haicku.net/cgi-bin/mailman/listinfo/%LIST`

- Unsubscribe Url (for majordomo at CERN, it is majordomo_interface.php?func=subscribe&list=%LIST&mailserver=listbox.server@cern.ch):

  `http://aoslox.haicku.net/cgi-bin/mailman/listinfo/%LIST`

- Archives URl:

  `http://aoslox.haicku.net/pipermail/%LIST/`

- Administrative Interface Url:

  `http://aoslox.haicku.net/cgi-bin/mailman/admin/%LIST`

10.3. Frontend services

10.3.1. Forum, tracker and news management

We are given the option to activate/deactivate four different trackers (support, bug, task, patch), a news system and a forum. Forum and patch tracker are deprecated features, for which support is not provided.

Later on, we are asked to set default posting restrictions according to user type (anonymous, logged-in, group member), and group member roles for news and item handling (manager, technician or both). These settings can later on be adjusted by project admins, but by setting suitable default values, we provide a template that will make the project fully operative without the need for further configuration.

10.4. Project menu settings

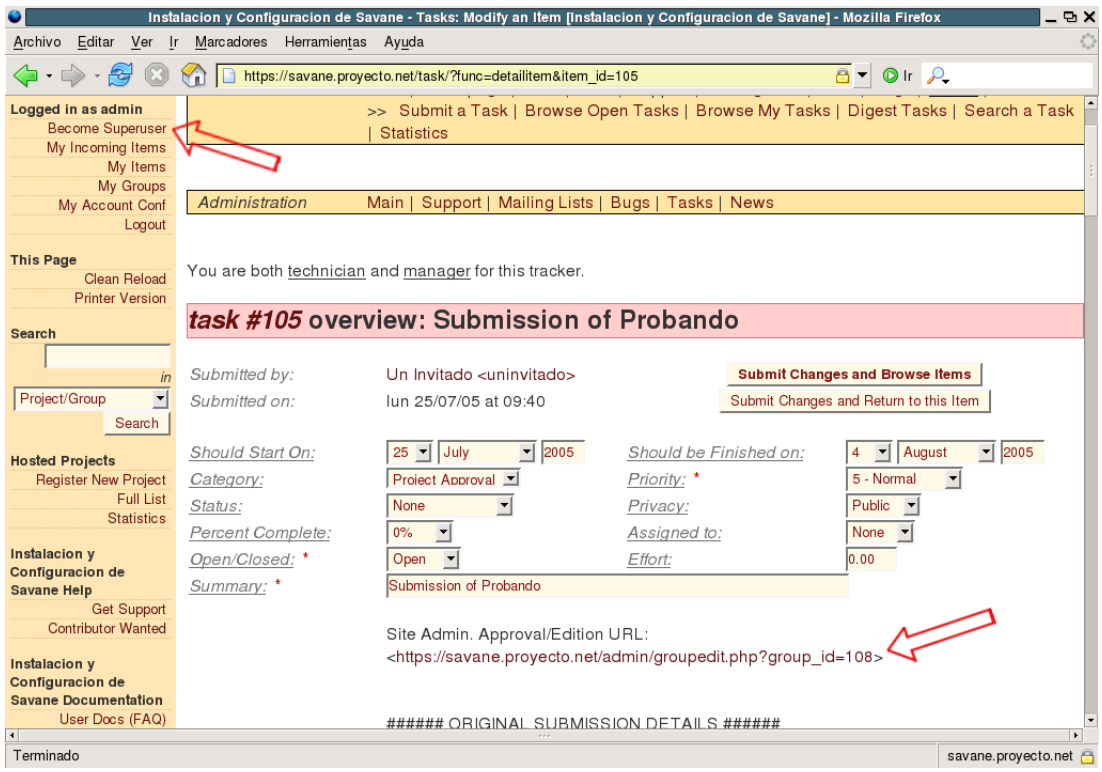
This section allows the server administrator to chose which service links can be altered by project administrators. Some of the options in this section might be dangerous, giving backend access, and the possibility to override much of the configuration already done by server admin. However, since several group types can be configured, you can allow different projects different level access (giving more autonomy to trusted projects, for example).

Chapter 11. Project approval

Another important superuser responsability is to approve/reject projects seeking hosting at the site. Project registration is not an automatic process: site administrator must check the user's submission before approving the project,and trigger its creation.

The process works as follows:

1. Once a registered user submits a project, a new task item under the Project aproval category is opened.
2. Site admin will be informed of this opened task, through the My Incoming Items section, appearing as a New and Unassigned Item.
3. However, to approve/reject it, admin must first become superuser, and then visit the "Approval/Edition" URL (see screenshot below).
4. If all is correct, superuser can then set the Status of the project to Active; update project settings (Update button- if it is not pressed, changes in settings will have no efect); and visit the "Trigger Project Creation" link. This will generate several backend actions in the next cron jobs, and send a mail to the user that submitted the project informing of its approval.
5. Admin, without the need of being superuser, can then return to the My Incoming Items section, and "close" the task.



Appendix A. GNU Free Documentation License

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the

copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

GNU FDL Modification Conditions

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the

Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you

also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Sample Invariant Sections list

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

Sample Invariant Sections list

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.