

SWAMP - Installguide

How to get the SWAMP running

1. System requirements

The following tools are needed and not included in the SWAMP distribution:

- Java \geq 1.4.2 SDK (<http://java.sun.com>)
- Apache Tomcat Servlet container (<http://jakarta.apache.org/tomcat> \geq 5.0.30, see the tomcat chapter for running on older versions)

Especially the RSS authentication part is currently set up to run with tomcat 5.5.

- Graphviz Tool for generating workflow graphs (<http://graphviz.org>)
- MySQL Database backend (<http://mysql.org>, tested with version \geq 4.0.18, don't use version 4.1.12, as it seems to have a bug (<http://bugs.mysql.com/bug.php?id=11084>))

SWAMP is running on both Linux and Windows. It is also possible to deploy on another servlet container than Apache Tomcat, but this has not been tested yet.

2. Install with SUSE RPM

Easy! ;-) Just download the latest SWAMP RPM from here (http://sourceforge.net/project/showfiles.php?group_id=68771) for your distribution and install with

```
yast -i swamp-*.rpm
```

YaST will solve the dependencies and install all needed packages for you. All needed files are installed now and you can proceed with setting up the database backend.

2.1. Installation on SLES 10

- Install the SLES 10 RPM

SWAMP - Installguide

- install graphviz RPM from opensuse: wget <http://download.opensuse.org/distribution/SL-10.1/inst-source/suse/i586/graphviz-2.6-22.i586.rpm>; rpm -ivh graphviz-2.6-22.i586.rpm;
- An Apparmor profile is available here (<https://svn.sourceforge.net/svnroot/swamp/trunk/conf/apparmor/tomcat-swamp.profile>)

3. Download the SWAMP source-code

To obtain the sources, either check them out via the subversion command:

```
svn co https://swamp.svn.sourceforge.net/svnroot/swamp/trunk/swamp swamp
```

or download the latest source tarball from

http://sourceforge.net/project/showfiles.php?group_id=68771 and untar the archive to your favourite location with:

```
tar -xvjf swamp-*.tar.bz2
```

You can use a common zip program for unpacking the archive on windows. The directory `swamp/` now contains the sources.

4. Install from source

Now you should create a host-specific config file to customize the build to your environment:

```
$ cd properties/  
$ cp host.properties `hostname`.properties  
$ vi `hostname`.properties
```

These settings should be self-documenting, really. They mainly set up the database environment and set the location of the tomcat installation. More general settings, that don't depend on the host you are developing on are contained in the `conf/` directory, but it should not be necessary to do changes here for a first installation. The build process makes use of the make-tool ant (<http://ant.apache.org>) which needs to be installed in the system to continue. To create the files needed for build out of your host-specific config-file and the template files, please run:

```
$ ant local-init
```

The next step is the creation of the swamp database, and generation of database dependant code. Thus we need a running MySQL instance and a valid user as described in the next section. The command:

```
ant torque-init
```

sets up the database tables and creates the Torque code for database access.

For the next step you need to have the msgfmt executable in your \$PATH for adding the translated strings from the .po files to the binary.

If everything went OK please call the wrapper script:

```
bin/install-swamp.sh
```

or on windows:

```
bin/install-swamp.bat
```

to compile and install the SWAMP server, the webSWAMP webapp and the SWAMP SOAP interface to tomcat. After starting/restarting tomcat your first SWAMP installation can be accessed by browsing to the URL:
<http://localhost:8080/webswamp/swamp>

5. Special notes for building on Windows

- Note that on windows systems you need to use "/" as a filesystem separator in the config files. (For example C:/temp)
- Ant can be downloaded from: <http://ant.apache.org/bindownload.cgi>. Please set your PATH variable to include the ant/bin dir.
- Please set the environment variables ANT_HOME and JAVA_HOME to point to your ant and Java installations.
- You need to have the msgfmt.exe in your \$PATH to be able to build the translation files. It can be downloaded from here:
<http://gnuwin32.sourceforge.net/packages/gettext.htm>

Maybe you also need the libiconv2.dll in your gettext\bin directory, it can be downloaded here: <http://gnuwin32.sourceforge.net/packages/libiconv.htm>

- To generate the graphs on Windows, download Graphviz here: http://www.graphviz.org/Download_windows.php and extend your PATH variable to include the graphviz/bin directory.

6. Set up the MySQL database backend

SWAMP needs a database backend to persist the applications state to disk. This means, each workflows data and status is stored in the db backend.

To make the database handle utf-8 characters properly you have to start it in utf8 mode by adding:

```
[mysqld]
default-character-set=utf8
```

to the /etc/my.cnf file.

6.1. Adding a SWAMP MySQL user

The easiest way to get it running is to use the MySQL root user for SWAMP, but we recommend you to create an extra user with restricted access rights due to security reasons. Nevertheless if you are fine with using the root user, you can skip this section, and can proceed with adding the credentials to the config file.

A new database user can be added with several GUI tools, or directly via the mysql shell. Enter the following command to start the shell:

```
mysql -u root -p
```

Now, we add the SWAMP user with access rights to the swamp db:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER
ON swamp.*
TO 'swamp'@'%' IDENTIFIED BY 'swamppwd',
'swamp'@'localhost' IDENTIFIED BY 'swamppwd';
```

This user is now able to log into the mysql server from any host (%) with the configured password.

If you are building SWAMP from source or developing it, the db user needs additional rights to create/delete the swamp db:

```
GRANT ALL ON *.* TO 'swamp'@'%', 'swamp'@'localhost' WITH GRANT OPTION;
```

Make sure to revoke these privileges on a production server!

6.2. Creating the SWAMP database

Since the created user has no privilege to create new databases, we need to create the SWAMP database:

```
CREATE DATABASE IF NOT EXISTS swamp;  
quit;
```

Now we can import the initial SWAMP tables: (when you build from source this will be done automatically during the "ant torque-init" command)

```
cd /srv/www/tomcat5/base/webapps/webswamp/WEB-INF/sql;  
mysql -u swamp -p swamp < swamp-schema.sql;  
mysql -u swamp -p swamp < swamp-security-schema.sql;  
mysql -u swamp -p swamp < turbine-schema.sql;  
mysql -u swamp -p swamp < swamp-scheduledjobs.sql;  
mysql -u swamp -p swamp < swamp-security.sql;
```

6.3. Add the values to the config file

The added SWAMP MySQL user and database name have to be added to the applications config file. If you installed from the RPM, this file resides at: `/srv/www/tomcat5/base/webapps/webswamp/WEB-INF/conf/Torque.properties`. If you are building from source, you should add the values to your `host.properties` file before building.

7. Setup SWAMP users

SWAMP uses the MySQL backend by default to store user information such as the users password, email-adress and so on. If you want to use another backend (eg. LDAP) for user authentication, please read the corresponding chapter in the developers manual.

Adding new users is not yet implemented in the webSWAMP frontend. This has to be done directly in the database. Easy GUIs for administrating MySQL databases are for example phpMyAdmin (<http://www.phpmyadmin.net/>) and MySQL Administrator (<http://www.mysql.com/products/tools/administrator/>). For security

SWAMP - Installguide

reasons, the users passwords are not stored in clear-text, but just a MD5 Hash of the password. A MD5 representation of the password string can for example be generated by the command:

```
perl -e 'use Digest::MD5 qw(md5_hex); print md5_hex("password");'
```

or

```
echo -n "password" | md5sum;
```

By default, there are already three accounts set up:

username: *mrswamp*, password: *swamp*

username: *swamp_admin*, password: *!swamp#*

username: *swamp_user*, password: *swamp*

Please change these values in a production environment.

When a user has logged in, the webSWAMP GUI provides a "Preferences" page where he can change his data including his password and preferred language.

8. Setup Apache Tomcat

The easiest way (on SuSE) is just to install the tomcat5 RPM, which can be done easily with YaST. On other distributions / operating systems, please use the package management tools or download the official distribution available from <http://jakarta.apache.org/tomcat> which should work just as well. We are developing on Tomcat 5.0.30, for older versions, please read the issues paragraph.

8.1. Tomcat issues:

SLES9 (tomcat 5.0.19):

- - Do a: `chown -R tomcat.tomcat /srv/www/tomcat/base/webapps/webswamp/` otherwise tomcat will not be able to write files
- - Remove: `/usr/share/tomcat/common/lib/commons-collections.jar` as it conflicts with the collections lib installed by SWAMP.

8.2. Setting up Tomcat:

The following instructions are only necessary if you want to do administrative tasks on the tomcat server itself. If you don't, just skip them.

Go to `$TOMCAT_BASE/conf`. (`$TOMCAT_HOME` is `/srv/www/tomcat5/base/conf` for the SuSE rpm). Edit the file `tomcat-users.xml` and make the following changes:

```
<role rolename="tomcat"/>
+ <role rolename="manager"/>
+ <role rolename="admin"/>
- <user username="tomcat" password="tomcat" roles="tomcat" />
+ <user username="tomcat" password="tomcat"
+ roles="tomcat,admin,manager"/>
<user username="role1" password="tomcat" roles="role1"/>
```

This means that the user "tomcat" with password "tomcat" (please use a sensible password here!) can now access tomcats manager and admin web-application.

8.3. Starting, restarting, and stopping Tomcat

For the SuSE rpm, issue as root:

```
# rctomcat start
# rctomcat restart
and
#rctomcat stop
```

You can set additional options by setting the variable `JAVA_OPTS` in `/etc/sysconfig/j2ee`, e.g.:

```
JAVA_OPTS="-server -Djava.net.preferIPv4Stack=true -Xmx128M"
```

On a manual installation, go to `$TOMCAT_HOME/bin` and call `startup.sh` and `shutdown.sh` at your leisure.

Check `http://localhost:8080/`. You should see the Tomcat default page there.

8.4. Reloading web applications:

Go to `http://localhost:8080/manager/html` and log in with the user `tomcat/tomcat` (or whatever you set up in the step before). You see a list of web applications Tomcat

has found in its webapp directory. If you rebuilt and re-installed a webapp, locate it in this list and click on the "reload" link.

To reload a webapp from the command line you can use:

```
curl -u <user>:<password>@http://<hostname>:8080/manager  
/html/reload?path=/webswamp
```

To avoid automatic re-deployment of the webapp on changes in config files, set `autoDeploy="false"` in the server.xml's `<Host>` element.

9. SWAMP config files

On an installed SWAMP system there are some config files that have to be configured for your environment.

9.1. conf/defaults

The main config file for the server.

In this file you can configure for example the used mailserver and authentication parameters.

9.2. conf/Torque.properties

Database connection parameters are configured here.

9.3. conf/TurbineResources.properties

Configuration for the web-frontend webSWAMP

Here you can set the hostname and port parameters so that links can get generated correctly.

10. The first run of SWAMP

To start SWAMP, please start the tomcat server with:

```
rctomcat5 start
```

After that, webSWAMP is available at the URL:
<http://<hostname>:8080/webswamp/swamp>. The default SWAMP installation includes two example workflows which will be described in this section.

10.1. Workflow: Example

This workflow is just a presentation of SWAMPs core features. It includes examples for branching and merging the workflow's thread, and contains examples for different kinds of possible tasks. All elements that were referenced in the adminguide in the "How to write a workflow template" section can be found here. That should provide you with a good insight on how the different elements actually work.

10.2. Workflow: Meeting

The meeting workflow is "real-world" example of a useful workflow. It helps to organise a meeting which includes notifying the participants of the meetings data and agenda, collecting the meeting minutes afterwards and send them to all attendees. The next section contains a detailed description of the meeting workflow and its steps.

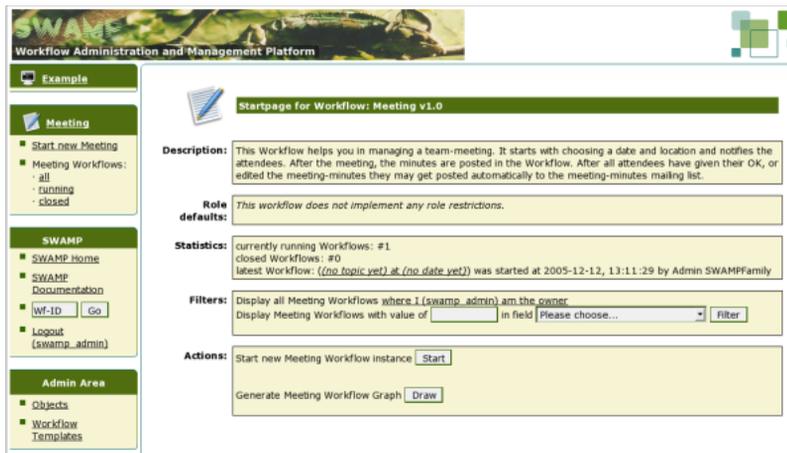
10.3. Workflow: minimal-test

This workflow contains the absolute minimum configuration for a valid workflow. It is basically included to test the framework, but it is also a good starting point for building your own workflow from scratch.

11. Example workflow: organizing a meeting

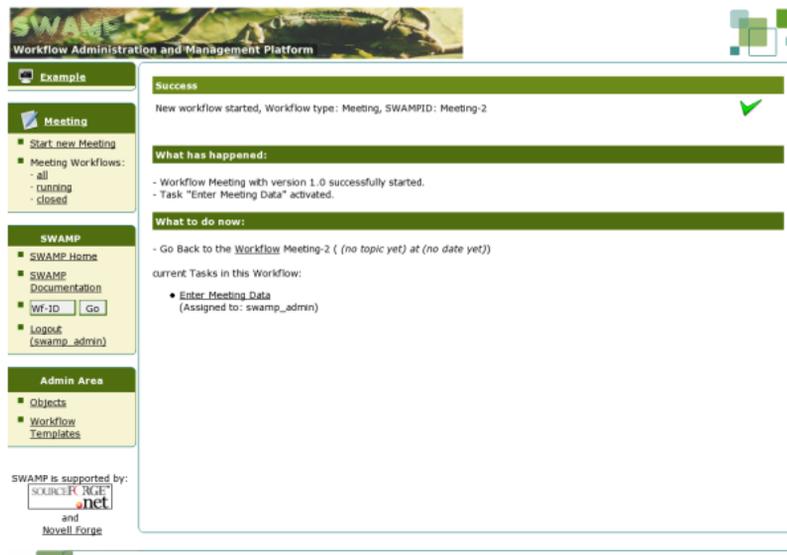
Now that you succeeded in installing SWAMP, its time to get a real workflow started. The SWAMP distribution contains an example workflow called "Meeting" that helps to organize a meeting and distribute the meetings result by email afterwards.

Please direct your browser to `<swamp-host>:8080/webswamp/swamp` to get to webSWAMPs login screen and login with the credentials that were described in the last chapter (`swamp_user:swamp` for example). After selecting the "Meeting" link on the left navigation bar the following start-screen of the meeting workflow is displayed:



This is the default workflow startscreen. It contains all general information of the meeting workflow and some statistical data, for example how many running instances of the meeting workflow exist in the system, and who has started the latest one. The page also contains links to start a new meeting workflow and to generate a graphical workflow graph.

So let's go and start a new meeting workflow by clicking on "Start". The following screen will show up:



The system tells you that it has successfully started a new meeting workflow, and that the first task of it has been activated. Because you started it, you are now the owner of that workflow, and got the first task "Enter meeting data" assigned. You can proceed to this task by clicking on the link, and will get this page presented:

Description of this Task:
Enter Meeting Data

Task ID	Description	Assigned Users	Workflow ID
2	Enter Meeting Data	swamp_admin	Meeting-2

Milestones:
 Initial data
 Minutes draft ready
 Minutes acknowledged and published

Meeting Dataset

Topic of the meeting: *

When will the meeting be?: *

Location of the meeting: *

Meetings Agenda: *

Involved people

Attendees of the meeting: (Type: Login) *

Who should take the meeting minutes?: (Type: Login) *

Fields with a (*) are mandatory.

This page contains your first task in the workflow. You have to fill in some fields to tell the workflow what the meeting is all about, for example the topic, date and agenda of the meeting. You have to enter a list of attendees and specify the username of the person who will take the meeting minutes. SWAMP expects a list of comma separated usernames that are known to the system. Just putting in email addresses is also allowed. Fields that are marked with a "*" in the last column are mandatory. You can try what happens when you leave out a mandatory field.

When you have completed your task, you will be redirected to a page that shows the results of your action. There have been some events been triggered in the background, for example a email notification has been sent out to all attendees containing the information you entered. Also, the person who should take the meeting minutes got a reminder to post the minutes in a SWAMP task as soon as the meeting has been held.

Show 2 of 2 workflows
 activated Systemfilters: Template = Meeting && workflow running &&

Workflow-Template: **Meeting** | Display Subworkflows: all | [configure columns] | [1] items per page: 30 | Set

ID	Description	Next Tasks
2	inttools weekly at 2005-12-12	• Enter Meeting Data (swamp_admin)
1	inttools weekly at 2005-12-19	• Enter Meeting Data (swamp_admin)

Workflow Colours: inactive | printable View | [1] items per page: 30 | Set

The last screen shows the workflow overview page the can be reached by the links

SWAMP - Installguide

"running", "closed" or "all" workflows from the navigation bar on the left.