OurMap Installation Guide

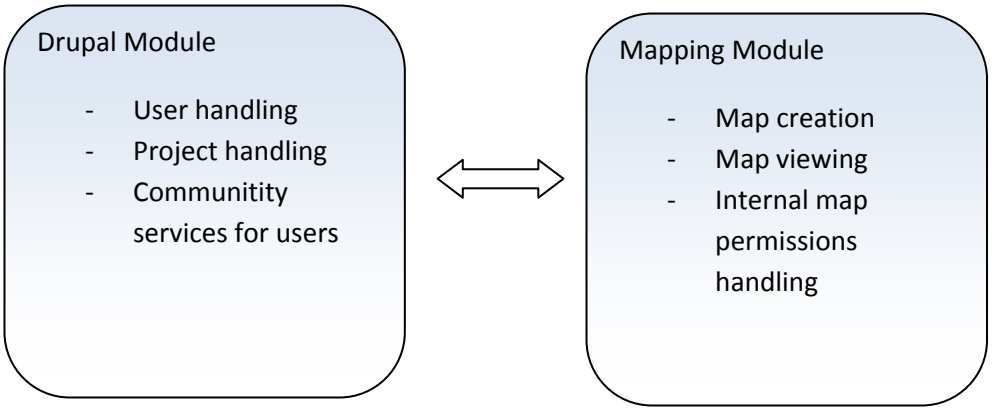
Version history

Version	Date	Author	Comments
1.0	Nov 12, 2010	Manuel Cuesta camilocuesta@hotmail.com	Initial draft

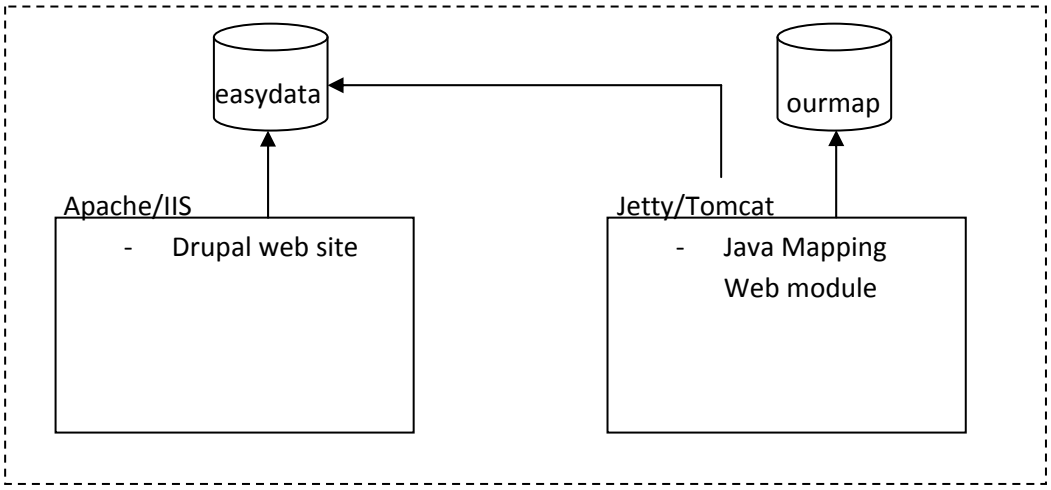
Contents

Overview

OurMap is a mapping application that relies on two modules: A drupal website, and a separate web application that will handle the mapping features:



Here you'll find the instructions of how to install the Drupal Module, as well as the Mapping Module. Since the Drupal Module will run on php, and the Mapping module on a Java, you will need two servers running on your machine, and two database for each server, as shown in the diagram:



Installation of the Mapping Module

0 Prerequisites

1. You need Java 6 or higher (http://www.java.com/en/download/manual.jsp)
2. You need Apache Ant

Once sucessfully installing Java and ant, you must be able to type "java –version" on the prompt and see the java version (version 1.6 in this case):

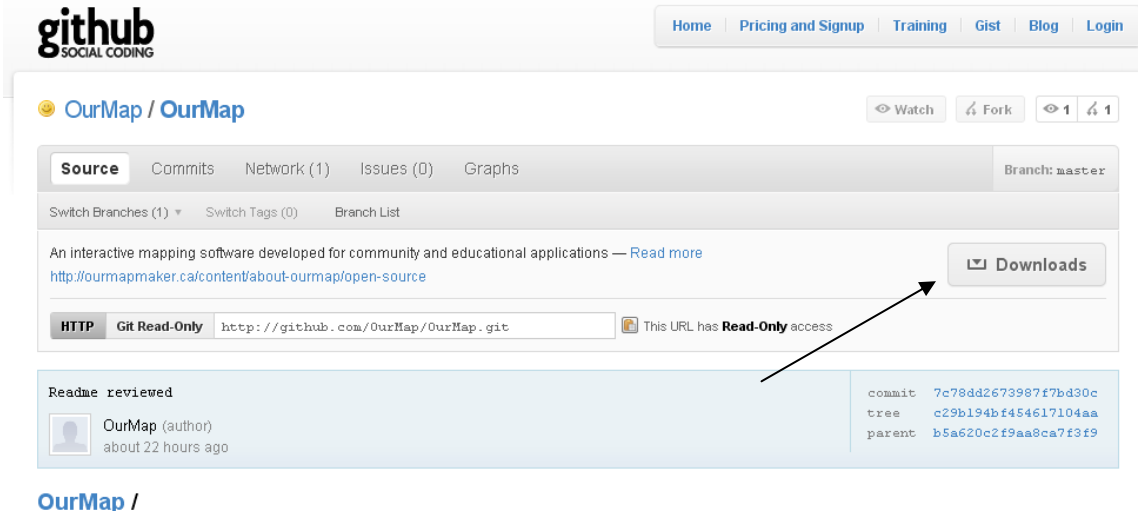
```
C:\>java -version
java version "1.6.0_07"
Java(TM) SE Runtime Environment (build 1.6.0_07-b06)
Java HotSpot(TM) Client VM (build 10.0-b23, mixed mode,
sharing)
```

And, typing "ant –version" must show you the respective ant version (version 1.7.1 in this case):

```
C:\>ant -version
Apache Ant version 1.7.1 compiled on June 27 2008
```

1 Get the sources

Download the last version of OurMap at http://github.com/OurMap/OurMap . Click on the "Downloads" button at the right of the page:



OurMap /

Unzip the folder at your desired location, which we are going to call the "OurMap installation folder".

2 Libraries needed

You need the libraries below to compile the sources, and copy all the respective .jar files to the “lib” folder of the OurMap installation folder.

- log4j

Version : 1.2.16 or higher
Tested with file: log4j-1.2.16.jar
Get it from here: http://logging.apache.org/log4j/1.2/download.html

- PostgresSQL JDBC Driver

Version : 8.2-504.jdbc2 or higher
Tested with file: postgresql-9.0-801.jdbc4.jar
Get it from here: http://jdbc.postgresql.org/download.html

Name : MySQL JDBC Driver
Version : 5.1.6 or higher
Tested with file : mysql-connector-java-5.1.13-bin.jar
Download link: http://www.mysql.com/downloads/connector/j/

- Struts 1.3.10

Version : 1.3.10
Download link: http://struts.apache.org/download.cgi#struts1310
You need only the “Library” files. There, the following jar files are need to compile the sources:
jstl-1.0.2.jar
oro-2.0.8.jar
antlr-2.7.2.jar
commons-logging-1.0.4.jar
commons-digester-1.8.jar
commons-chain-1.2.jar
commons-validator-1.3.1.jar
bsf-2.3.0.jar
standard-1.0.6.jar
commons-beanutils-1.8.0.jar
commons-fileupload-1.1.1.jar
commons-io-1.1.jar
struts-core-1.3.10.jar
struts-taglib-1.3.10.jar
struts-tiles-1.3.10.jar
struts-el-1.3.10.jar
struts-extras-1.3.10.jar
struts-faces-1.3.10.jar

struts-mailreader-dao-1.3.10.jar
struts-scripting-1.3.10.jar

- Servlet and JSP library

Tested with files: “servlet-api” , and “jsp-api.jar”
Get it from Apache: http://tomcat.apache.org

Thus, the following list of 27 jar files is our suggestion to compile the whole application:

antlr-2.7.2.jar
bsf-2.3.0.jar
commons-beanutils-1.8.0.jar
commons-chain-1.2.jar
commons-digester-1.8.jar
commons-fileupload-1.1.1.jar
commons-io-1.1.jar
commons-logging-1.0.4.jar
commons-validator-1.3.1.jar
cuestadao.jar
CuestaLib.jar
jsp-api.jar
jstl-1.0.2.jar
log4j-1.2.16.jar
mysql-connector-java-5.1.13-bin.jar
oro-2.0.8.jar
postgresql-9.0-801.jdbc4.jar
servlet-api.jar
standard-1.0.6.jar
struts-core-1.3.10.jar
struts-el-1.3.10.jar
struts-extras-1.3.10.jar
struts-faces-1.3.10.jar
struts-mailreader-dao-1.3.10.jar
struts-scripting-1.3.10.jar
struts-taglib-1.3.10.jar
struts-tiles-1.3.10.jar

3 Compiling the sources

Locate yourself at the OurMap installation folder created in step 1.

You need to compile the OurMap module first, and then the “OurMapWeb” module.
Compile the “OurMap” module using Ant with the following command:

```
>ant -f build-ourmap.xml
```

On execution of this command, the “BUILD SUCCESSFUL” message must appear to indicate that there were no errors.
The folders “classes” and “dist” are created, and the OurMap.jar file is created and placed inside the “dist” folder.

To compile the “OurMapWeb” module, run the following command:

```
>ant -f build-ourmapweb.xml build
```

Make sure the message “BUILD SUCCESFUL” appears here as well.
The file “OurMapWeb.war” is created and placed in the “dist” folder. If you want the .war file to be copied to your desired deploy folder, you need to set the “webapps.dir” parameter in the ant.properties file:

```
# Deploy dir for the war file
webapps.dir = temp
```

Replace “temp” with the absolute path of your desired deploy folder, without the trailing slash “/”. Then, to execute the copying of the file, type this command:

```
>ant -f build-ourmapweb.xml
```

4 Configuring the application

Take a look at the ourmap.properties file

```
db_url = jdbc:postgresql://localhost/ourmap
db_user = your postgres user
db_password = xxxxxx
db_driver = org.postgresql.Driver
googlekey=your google key
drupal_db_url = jdbc:mysql://localhost/easydata
drupal_db_user = your_mysql_user
drupal_db_password = xxxxxx
drupal_db_driver = com.mysql.jdbc.Driver
filesFolder = http://yourserver/media
baseDir = /yourpath/media
```

db_url : This is the url for the mapping database. The default scenario is that you install the mapping database on your local host, under the name “ourmap”, and it is a postgres database. Otherwise, you need to change this url.
db_user: The postgres mapping database user. You need to provide your user here.
db_password : The postgres mappng database password.You need to provide your password here.
db_driver : Driver class for the mapping database. The default is postgres.
googlekey : the key for showing your maps in your server with the Google API version 2. Replace this value with your key.
drupal_db_url : The url for the drupal database, which is supporting the drupal site. The default scenario is a mysql database for the drupal site, named “easydata”.
drupal_db_user : The user for the drupal database. You need to provide your user here.

drupal_db_password : The password for the drupal database. You need to provide your password here.
drupal_db_driver : Driver class for the drupal database. The default is mysql.
filesFolder : Path of the virtual directory for your media files. Without the trailing slash.
baseDir : Absolute path of the directory on your disk where the media files are located. This directory should be visible as a virtual directory with the "filesFolder" above.

For a default installation you need to:

- provide the user of the mapping database in the db_user property.
- provide the password for that user.
- provide your google key in the googlekey property.
- provide your user for the drupal database in the drupal_db_user property.
- provide your password for that user.
- provide the path of the virtual directory of the users' media files, without the trailing slash.
- provide the path of the directory in disk where the users' media files are going to be located, without the trailing slash.
- and lastly, place the ourmap.properties file on the working directory of the java server. For example, if you are using Tomcat, place it on the "bin" directory. Or, if you are using Jetty, place it directly on the directory where Jetty is installed.

5 Setting up the mapping database

The suggested scenario for installation of the mapping database is to have your "ourmap" database in a postgres server. In this case, you need to follow this steps:

- Create a database named "ourmap"
- Run the script "ourmap_initial_101028.sql" on that database. You'll find the script at the OurMap installation folder.

6 Setting up the virtual directory

The installation of the virtual directory depends greatly on your choice for server. On a Jetty server for example, all you have to do is create a xml file on the "contexts" folder that looks like this:

```
<Configure class="org.mortbay.jetty.handler.ContextHandler">
   <Set name="contextPath">/usermedia</Set>
   <Set name="resourceBase">c:/usermedia</Set>
   <Call name="addHandler">
       <Arg>
           <New
class="org.mortbay.jetty.handler.ResourceHandler"/>
       </Arg>
   </Call>
</Configure>
```

Given that you want to name your virtual directory as "usermedia" and that actual folder is located at "c:/usermedia" . The xml file can have any name.

Then you must set the ourmap.properties file accordingly:

```
filesFolder = http://yourserver/usermedia
baseDir = c:/usermedia
```

7 Icons

The installation package, and the provided mapping database (on the file ourmap_initial_101028.sql), has 22 built-in icons, to give you a head start with your mapping application.

Copy the folder "icons" which is inside "usermedia" on the OurMap installation folder, and place it on the folder of the "baseDir" property on the ourmap.properties file (see the above step).

In this case, if you are using "c:/usermedia" to store the users' media files, then you must copy the icons folder to that location:
c:/usermedia/icons

Installation of the Drupal Site

by Jeff Bolingbroke (jeffrick@hotmail.com)

Re-creating the OurMap system somewhere new should not be seen as a daunting task, but there are some important things to remember when re-establishing it somewhere new. The following is a step-by-step guide to cloning the site.

*Note: I should mention that I am not in any way an expert at this part of site building. If you have any problems or difficulties, I'd suggest starting with the drupal.org website – as Drupal is open source, there's lots of documentation on how to do most things youwill want to do, plus a very good help forum to peruse. I found talking to IT professionals often helped as well)

Step 1: Download the files and save them on your server

Your server machine should be set up with a http web server. We used Apache2 running PHP 5.2
We have provided all of the necessary files to get you started. Simply unzip the archive to a web-accessible area of your server.

*Note: One thing that can go wrong here is that when you copy the files, you don't actually copy hidden files. The .htaccess file is very important, so make sure when your copy ends up in its new location, that hidden files match the original.

Step 2: Re-create the Database

We have included a file called ourmap.sql. This file contains all of the database information. You'll need to create a new database on your server and import this file.

If you don't have a MySql database engine yet, you'll need to install one. In this link you can find detailed information on how to install your own MySql instance:
http://dev.mysql.com/doc/refman/5.1/en/installing.html

Create a database on your mysql with the name "easydata" . To create the database, log in on your mysql database engine, and enter the following command:

```
> create database easydata ;
```

Then, log off, and restore the database using the "easydata_101005.sql" file. You can find this file on the OurMap installation folder. Use this command from your operating system prompt, to restore the database:

```
>mysql -u YOUR USER -pYOUR PASSWORD easydata <
easydata_101005.sql
```

Step 3: Modify the Settings.php file:

In the settings.php (located in the /easymap/sites/default directory) file, there are a couple of key lines that you'll need to modify to reflect the location of the URL and database of the site.

A) The first is the **Database Settings** (line 92)

It looks like this by default (i've replaced the password with xxxxx):

```
$db_url = 'mysql://YOUR USER:xxxxxx@localhost/easydata';
$db_prefix = '';
```

You need to make the change to reflect the new way for the site to access the database. Format should be: mysql://username:password@serveraddress/database name

B) The **Base URL** code should also be modified to reflect the new site. The default setting is:

```
$base_url = 'http://localhost';
```

You want to change the address to your new URL, like this:

```
$base_url = 'http://mynewURL.com';
```

Make sure to save your changes, and then set the permissions for the settings.php file in such a way that they can't be modified by anyone but the administrators.

At the bottom of the file, you'll find the settings of the integration with the mapping site:

```
$mapsiteurl  = 'http://localhost:8080/map/' ;
$map_db = 'ourmap' ;
$map_host = 'localhost' ;
$map_port = 5432 ;
$map_user =  'YOUR POSTGRES USER' ;
$map_pwd = 'XXXXXX' ;
```

mapsiteurl: This is the url of the context for the mapping site. For example, if you use Jetty, the context in which the java map application is deployed, must match this value. map_db, map_host, map_port, map_user and map_pwd: This is the configuration to access the postgres database.

If you use the recommended configuration, you only have to change the following properties:

- Map_user : Must match the “db_user” property on step 4
- Map_pwd: Must match the “db_password” property on step 4

If you need to change the map_db, map_host, or the map_port properties, the values must match the corresponding “db_url” property on step 4.

Step 4: Install necessary programs on the server

OurMap uses several small programs on the current server. The clones version will be looking for these on the new server, so if they are not installed, it would be wise to do so:

- GD2 Image Library

Step 5: Make sure the permissions are correct:

The /sites/default/files/ directory, and everything inside it, needs to have the permissions set in such a way so as to allow the Drupal site to write and read things inside it.

Step 6: Clean Urls

You’ll need to tell Drupal to use “clean Urls”, follow this link for instructions of how to do it on apache servers: http://drupal.org/getting-started/clean-urls , and this link for instructions on IIS : http://drupal.org/node/3854

Step 7: Try it out!

Navigate to the base URL you've indicated in the settings.php file and see what happens. With any luck, you'll see the site. If not, there will be some kind of error message, which wil provide you with a clue as to what has gone wrong. Troubleshooting can be a little time-consuming, but you should be able to pinpoint the problem with a little reading on drupal.org

Note: We noticed some real problems occurred when we moved the site, which had been created on a server that used PHP 5.2 and tried to use it on a server with PHP 5.3. In the end, we installed 5.2.4-2ubuntu5.10 on the new server and the problems went away.