



FOSS4G 2007 Lab L-06

Practical Introduction to MapStorer

**Verena Diewald, Arnulf Christl,
WhereGroup**



Agenda

- Introduction
- History
- Installation and configuration
- Functionality and handling
- Future prospects



What is MapStorer ?

- MapStorer is a Free Software / OpenSource project
- MapStorer is a database supported system for managing MapServer projects
- Mapping projects can be created and edited via an easy-to-use web frontend
- MapServer configuration files (mapfiles) are created on the fly
- Prompt preview of created and edited mapping projects
- Reduces manual editing of text files
- Organizes mapfiles
- see also <http://www.mapstorer.org>



History of MapStorer

- 2004-01 Conception of MapStorer is developed
- 2004-03-19 A first "in-house" version is completed
- 2004-12-08 first release under the GNU GPL license
- 2005-02-21 MapStorer is registered on SourceForge and available for download there (<http://sourceforge.net/projects/mapstorer/>)
- 2007-06-29 MapStorer version 0.8 is available for download
- 2007-09-19 Mapfile import milestone is reached



What you need for running MapStorer

- Web server
- MapServer version 4.8 or newer
- Database MySQL or PostgreSQL
- PHP installed and configured in web server
- PEAR (PHP Extension and Application Repository)



How to start with MapStorer

- download MapStorer from <http://sourceforge.net/projects/mapstorer/>
- copy directory "mapstorer" to the document root of your webserver
- configure database connection

```
// MySQL configuration

/* MySQL
  define('DBHOST',      'localhost');
  define('DBDRIVER',    'mysql');
  define('DBUSER',      'root');
  define('DBPASSWORD',  '');
  define('DBPORT',      '3306');
  define('DBNAME',      'mapstorer');
/**/

// PostgreSQL configuration

/**/ PgSql
  define('DBHOST',      'localhost');
  define('DBDRIVER',    'pgsql');
  define('DBUSER',      'postgres');
  define('DBPASSWORD',  '');
  define('DBPORT',      '5432');
  define('DBNAME',      'mapstorer');
/**/
```



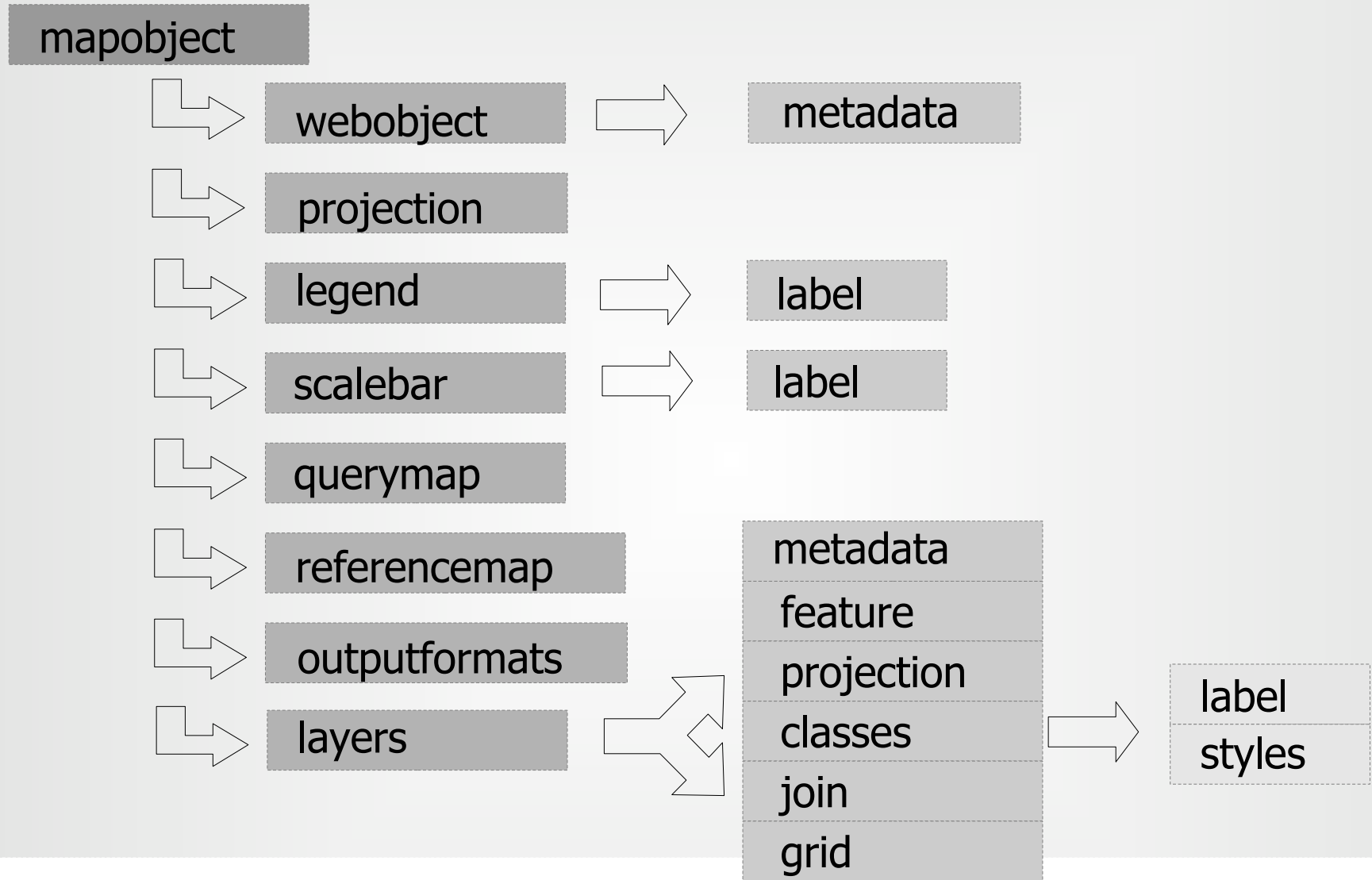
Configuration of MapStorer

- Point your browser to Mapstorer (eg. <http://212.79.172.185/mapstorer09/>)
- Log in as user "root" and password "root"
- Select "Global configuration" on the navigation form (right frame) and edit the given parameters

Configuration section: mapstorer_core	
*IMAGEPATH	<input type="text" value="/var/www/mapstorer/ms_tmp/"/>
*IMAGEURL	<input type="text" value="http://localhost/mapstorer/ms_tmp/"/>
*Template path	<input type="text" value="/var/www/mapstorer/preview/preview.html"/>
*Mapserver CGI URL	<input type="text" value="http://localhost/cgi-bin/mapserv"/>
*Preview mapfile path	<input type="text" value="/var/www/mapstorer/preview/mapfiles/"/>
*Preview data path	<input type="text" value="/var/www/mapstorer/preview/data/"/>
*Export path	<input type="text" value="/var/www/mapstorer/export/"/>
GDAL tile index executable	<input type="text" value="/usr/bin/gdaltindex"/>
Create tile indexes?	<input type="radio"/> Yes <input checked="" type="radio"/> No
*Mapstorer URL	<input type="text" value="http://localhost/mapstorer/"/>



Data model





Functionality and handling

1. Creating a new mapfile
2. Adding layers to a mapfile from different data sources
3. Creating layer from postgis database
4. Creating classes for a layer
5. Re-using layers from any mapfile in any other mapfile (combine mapfiles)
6. Exporting mapfile as plain text
7. Importing mapfile into MapStorer



1. Creating a new mapfile

- **Button** "New mapfile" on navigation site
- Fill in `NAME`, `EXTENT` (should be `-180,-90,180,90`), `IMAGECOLOR` (use the color editor), `UNITS` (should be `DD`) and `PROJECTION` (`epsg:4326`)
- **Use button** "save" to write new mapfile in db and switch to mapobject perspective



2. Adding layers to a mapfile

- Go to tree menu to "layers" and push button "Add new layer"
- Select data source
- Add raster layer (DATA: „wsiearth.tif“)
- Add shape file layer (DATA: „bccityg.dxf_points.shp“, STATUS: ON, TYPE: POINT, CLASSITEM: ID) --> Don't see anything???
See next step...
- Select created shape file layer from layers list, go to created default class „class1“ (click on class name), go to edit mode and push button „get symbol“, choose a symbol, set SIZE to 10 and save changes, close window and preview mapfile again



3. Creating layer from postgis data

- Go to main panel using the button near tree menu
- Push button „Postgis Plugin“
- Use link „Connection storage“ from link list on the top of page
- Test stored connection to db „canada_demo“ if connection works fine, close this window
- Add new layer (source: database) to the working mapfile
- Call Postgis plugin clicking the Postgis image
- Choose connection to database „canada_demo“
- Take geometry layer „bound_p“ and go back to form
- Set STATUS DEFAULT and CLASSITEM „stateabb“, choose colors **and push „Save and preview“**
- Sort layers in layers list in right direction (1. Raster image layer, 2. Polygon layer from database, 3. Point layer from shape file) to see all created layers and see preview again



4. Creating classes for a layer

- Select created database layer from layers list
- Push button "Add new class" in class list
- Insert necessary class attributes (NAME: British Columbia, EXPRESSION: CA-BC, COLOR) and save class
- Close window and sort classes in class list in right direction
- Preview mapfile and see new created class



5. Re-using layers from mapfile in other mapfile

- Go to navigation site and push button "Combine mapfile"
- Copy one complete mapfile to a new one with "Duplicate existing mapfile" function
- Have a look at duplicated map in preview



6. Exporting mapfile as plain text

- Go to navigation site and push button "Export"
- Select mapfile to export from list
- See exported mapfile with original parameters instead of local pathes (for use on other servers)



7. Importing mapfile into MapStorer

- Go to navigation site and push button "Mapfile Import Plugin"
- Go to „Import Mapfile“
- Insert name for import mapfile
- Go to your file directory and browse for the mapfile to import
- Start import
- Select imported mapfile from treemenu, go to mapobject and preview the mapfile



Planned features

- Connection to Mapbender
- Symbol editor to create and edit symbols
- SLD editor to generate and edit user-defined symbols as styled layer descriptors
- Create mapfile history and versions
- Help system (online manual, links from form fields to mapserver mapfile online syntax help)
- Multi-language support, Internationalisation: Easily switch languages in mapfiles and/or complete frontend
- ...



Thank you for your attention!

For further information please contact:

Verena Diewald

email: verena.diewald@wherogroup.com

<http://www.wherogroup.com/>

This document is dual licensed:

GNU FDL license 1.2 (<http://www.gnu.org/licenses/fdl.html>)

CreativeCommons Attribution-ShareAlike 2.5 license (<http://creativecommons.org/licenses/by-sa/2.5/legalcode>)