Hidden features and useful tips for UMN MapServer powerusers

09.09.2010, FOSS4G 2010, Barcelona, Spain
Agenda

• What is UMN MapServer?
• Mapfile-Hacking
• MapServer-based Services with GMAP and VE
• Cartodiagrams
• Performance-Tuning
• Debugging
• What's next?
What is UMN MapServer?

• C-based Mapping-platform for creating spatial data and interactive mapping-components for the internet

• MapServer is NO complete GIS and it will not be in the future.

• Basically it's a rendering-software for creating maps

• MIT-style license, OSGeo-project

• platform independent (Windows, Linux, MacOSX)

• Supports vector- and raster formats as well
  – SHP, TIFF, GDAL/OGR, Oracle Spatial, Arc SDE, PostGIS and many more
How does it work
The Mapfile

• Configuration file for MapServer

• Editable in your beloved text editor

• Must end with .map

• Usage of INCLUDEs

• Preprocessing

• Versioning

---

The code snippet in the image shows a sample Mapfile configuration for MapServer.
Mapfile-Hacking

• Preprocessing
  – Using any sort of preprocessors like cpp
  – Define styling using 'themes'
  – Usage of include-templates -> result is one mapfile (good for debugging)
  – split your mapfile in logical groups and INCLUDE

• Versioning
  – In combination with preprocessing easily maintainable
  – Complex projects make this a must!
  – Need I say more on versioning?
Mapfile-Hacking

- Projection-block without EPSG-reference; add PROJ.4-string directly in your mapfile (it's fixed, not really needed any more)
- Use [MIN|MAX]SCALEDENOM-parameters
- Sort your CLASSes: frequently used classes on top!
- Simplify expressions
- Symbolset/Fontset: add only those fonts/symbols which are really used!
Mapfile-Hacking

- Use optimized output-FORMATOPTIONs (depending on your data):
  - QUANTIZE_FORCE=on
  - QUANTIZE_COLORS=256
  - QUANTIZE_NEW=on

- Hide layers from GetCaps which are useless there (e.g. Attributions, tileindex-layers, group-members) [currently requires patching, see #1952 for details]:
  'OWS_HIDDEN_LAYER' 'TRUE'
GMAP and VE with UMN

• Requirements:
  – MapServer compiled with PROJ.4-support
  – Every layer needs a PROJECTION-block
  – Spherical-Mercator-Projection:

```plaintext
+proj=merc +a=6378137 +b=6378137 +lat_ts=0.0 +lon_0=0.0 +x_0=0.0 +y_0=0 +k=1.0 +units=m +nadgrids=@null +no_defs
```

• New parameters:
  – mode=tile
  – tilemode=[gmap|ve]
  – tile={x+y+z}|{10231}

• example: file://./beispiele/gmaps.html
GMAP-configuration

```
<!DOCTYPE html>
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="content.type" content="text/html; charset=utf-8"/>
<title>Google/MapServer Tile Example</title>
<script src="http://maps.google.com/maps?file=api&v=2&key=[YOUR KEY HERE]"
  type="text/javascript"></script>
<script type="text/javascript">
function load() {
  if (GBrowserIsCompatible()) {
    var urlTemplate = 'http://demo.intevation.de/cgi-bin/frida-wms2?';
    urlTemplate += 'layers=gewaesser sehensverdienstige strassen2&';
    urlTemplate += 'mode=tile&';
    urlTemplate += 'tilemode=gmap&';
    urlTemplate += 'tile={X}://{Y}://{Z}';
    var myLayer = new GTileLayer(null, 0, 18,
                           tileUrlTemplate: urlTemplate,
                           isPng: true,
                           opacity: 0.5);)
    var map = new GMap2(document.getElementById("map"));
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());
    map.setCenter(new GLatLng(52.27, 8.04), 10);
    map.addOverlay(new GTileLayerOverlay(myLayer));
  }
}
</script>
<body onunload="load()" onunload="GUnload()">
  <div id="map" style="width: 500px; height: 500px"></div>
</body>
</html>
```
Cartodiagnostics

- Bar- and piechart-diagrams are supported for AGG and GD-Rendering
- New Layer-Type: TYPE 'CHART'
- Configuration is done with PROCESSING-directives inside the LAYER-block

```xml
<CLASS
  NAME  "Population Age 0-19"
  STYLE
    SIZE [1006]
    COLOR 255 244 237
  END

<CLASS
  NAME  "Population Age 20-39"
  STYLE
    SIZE [1007]
    COLOR 255 217 191
  END

<CLASS
  NAME  "Population Age 40-59"
  STYLE
    SIZE [1008]
    COLOR 255 186 140
  END

END>
```
Performance-Tuning

- Optimize/minimize the EPSG-table to the really needed entries your app supports
- Set WMS_EXTENT manually for your PostGIS-layers
- Use CLOSE_CONNECTION=defer
- Use Fast-CGI
- Preprocess and prepare your rasterdata as
  - Uncompressed, Tiled, With overviews
- Use good/clever scalehints for your layers, add eventualy overviews
Debugging

• Set Mapfile-parameters:
  - DEBUG [level]
  - CONFIG „MS_ERRORFILE“ „/tmp/ms_errors.log“

• Use shp2img:
  - shp2img -m <mapfile> -l <layer> -o <output> -all_debug 10

• Set Environment variables:
  - CPL_DEBUG=ON: GDAL/OGR-debugging-infos
  - PROJ_DEBUG=ON: PROJ4-debugging-infos
  - MS_ERRORFILE=/tmp/file.txt: save debug-messages here
Debugging

- Commandline-debugging of CGI-mapserv:

  # export needed environment variables
  export MS_MAPFILE=/path/to/mapfile.map
  export QUERY_STRING="request=GetCapabilities\&Service=WMS&Version=1.1.1"
  export REQUEST_METHOD=GET

  # run the CGI-process (-nh means no header)
  /path/to/cgi-bin/mapserv -nh

- Alternativly use cURL for sending requests, especially for POST-WFS-calls
Where do I get help?

• OSGeo Infrastructure
  – Mailinglists (user/dev)
  – Trac (browse the tickets, there a some treasures included [as patches]).
  – SVN
  – IRC (#mapserver on freenode)
  – Wiki

• Community-conferences (e.g. FOSS4G, German FOSSGIS)

• Commercial partners
What's next?

- Release 6.0 will (most probably) have
  - Pluggable rendering engines
  - Tile cache support
  - Output formats: GeoJSON, KML, GML3
  - Lots of minor stuff fixed
- More from the 'real' devs in the room!
Thank you...

Intevation GmbH
Stephan Holl
<stephan.holl@intevation.de>
Neuer Graben 17
49074 Osnabrück
+49 (0)541 – 335083 663