



MapServer *#ProTips*

MapServer for Power Users

A collection of tips to streamline your mapping workflow

09/17/15



#Protip: Discover performance issues, problems, etc with DEBUG

09/17/15



DEBUGGING - Part 1

- set DEBUG level at MAP and/or LAYER level
- can see GDAL/OGR debug info with CPL_DEBUG

```
MAP
```

```
...
```

```
CONFIG "CPL_DEBUG" "ON"
```

```
...
```

```
LAYER
```

```
...
```

```
END
```

```
END
```

- use shp2img command line utility with
‘-all_debug’ switch maximum debug info

09/17/15

DEBUGGING - Part 2

- can execute problem query at the command line (and avoid using Apache/Web server)

`mapserv -nh "QUERY_STRING="`

- <http://www.mapserver.org/optimization/debugging.html>

Debugging - Part 3

- Can debug to various locations
 - Web server log (default)
 - External log file (MS_ERRORFILE)
- Really advanced debugging (gdb)
 - Useful (but don't run production) (-DCMAKE_BUILD_TYPE=DEBUG)
 - you can set breakpoints in the code and invoke via web and inspect/change variables
 - can get backtraces (bt) to show developers where an error occurred

Debugging - Part 4

- Windows build debugging
 - sub-libraries must be built with debug symbols (such as GDAL, Apache)
 - check dlls loaded through DependencyWalker and ProcessExplorer (Sysinternals)
 - often issues of C++ redistributables

Debugging - Part 5

- DEBUG 2 isn't just about finding problems, it's also a way to track performance, provides layer timing (at the server)
- We (COE) store base performance times for important layers so we can have a baseline to refer to when users report issues and to evaluate new versions

#Protip: Secure OGC Services by IP

09/17/15



Securing OGC Services by IP

- Can be used at Map or Layer level
 - But most useful at layer level
 - Supports external files, address ranges, ipv6
- Can allow or deny by IP
 - can be used by service or all protocols
 - wfs_allowed_ip_list, ows_denied_ip_list
- Need to block cgi access when using this
 - "ms_enable_modes" "!" (block all cgi access)
 - "ms_enable_modes" "!* LEGEND" (allow legend call)

#Protip: Use of PROJECTION AUTO

09/17/15

PROJECTION AUTO

- part of MapServer since “early times”
 - largely under-used
- vector layers through CONNECTIONTYPE OGR
 - including shapefiles
- raster layers require no other config (handled through GDAL)
 - doesn't work with external world files

PROJECTION AUTO - Vector Example

LAYER

...

CONNECTIONTYPE OGR

CONNECTION "sqlite.db"

DATA "roads"

PROJECTION

AUTO

END

..

END

09/17/15

PROJECTION AUTO - Raster Example

LAYER

...

TYPE RASTER

DATA raster.tif

PROJECTION

AUTO

END

END

09/17/15

#Protip: Use WFS 2.0

09/17/15



WFS 2.0

- Paging is now default (MS WFS 1.1.x extension)
- Sorting supported (shapefiles via ogr)
- Time Support
- Stored Query Support
 - Defined server side
 - Supports stored queries with parameters
 - No CreateStoredQuery, DropStoredQuery
- Base for INSPIRE Download Services

09/17/15

WFS Examples

Time period queries:

```
<Filter><During><ValueReference>COLLECT_DATE</ValueReference><TimePeriod>  
  <beginPosition>2010-11-29T00:00:00-07:00</beginPosition><endPosition>2014-08-25T00:00:00-07:00</endPosition>  
</TimePeriod></During><SortBy><SortProperty><ValueReference>COLLECT_DATE</ValueReference><SortOrder>DESC  
</SortOrder></SortProperty><SortBy></Filter>
```

Define a stored query

```
"wfs_storedqueries" "mystoredquery"
```

```
"wfs_mystoredquery_filedef" "/data/mystoredquery.xml"
```

```
<Parameter name="datestart" type="xs:dateTime"/><Parameter name="dateend" type="xs:dateTime"/>  
<Filter><During><ValueReference>COLLECT_DATE</ValueReference><TimePeriod>  
  <beginPosition>${datestart}</beginPosition><endPosition>${dateend}</endPosition></TimePeriod></During></Filter>
```

Call a stored query:

```
&STOREDQUERY_ID=mystoredquery&DATESTART=2010-11-29T00:00:00-07:00&DATEEND=2014-08-25T00:00:00-07:00
```

09/17/15

#WFS Protip: (rfc-91)

- Major changes for ogr/ database users, attribute and spatial operations pushed to database
- Significant performance improvements if anything other than BBOX filters
- Uses MapServer filter syntax or `PROCESSING="NATIVE_FILTER=sql syntax"` or embedded in DATA statement

#Protip: HTML based Legends

09/17/15



HTML Legends

- An alternative to img based legends
- CGI only (incompatible with GetLegendGraphic)
- Template based generation
- Add TEMPLATE <path/to/template> in MapFile Legend Section

HTML Legend examples

JSON formatted:

```
[leg_class_html opt_flag=0]
  {"name": "[leg_class_name]", "icon": "[leg_icon]", "mykey" : "myvalue" },
[/leg_class_html]
```

HTML formatted:

```
[leg_class_html opt_flag=0]
  <img src=[leg_icon]> [leg_class_name]<br>
[/leg_class_html]
```

#Protip: Content Dependent Legends

09/17/15



Content Dependent Legends

- Works in OWS GetLegendGraphic mode or CGI mode=maplegend/maplegendicon
- requires extra parameters
 - BBOX, WIDTH, HEIGHT, SRS, LAYERS (not LAYER)
- Will do more work as it needs to query data to see if the data exists at that extent/scale

#Protip: Advanced Blending Modes

09/17/15



Blending Modes

- 7.0 introduces new LAYER->COMPOSITE blocks
- replaces older OPACITY use
- new COMPOP parameter
- modes include: contrast, darken, difference, multiply, overlay,...

Blending Modes

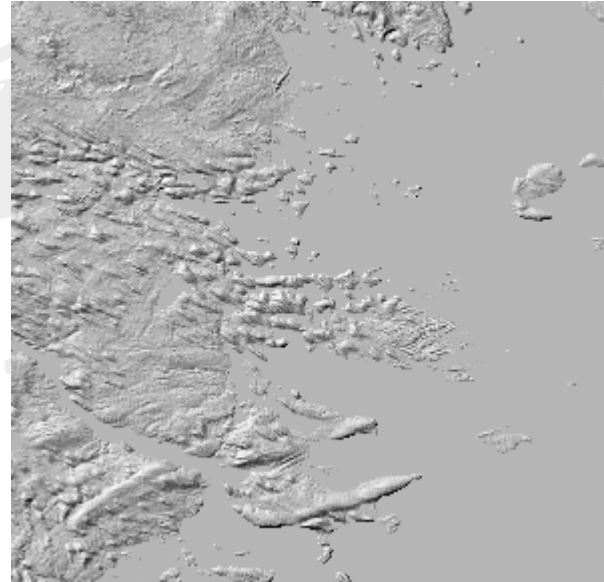
LAYER

..

COMPOSITE
OPACITY 70
END

..

END



09/17/15

Thank you for using MapServer!



@mapserving
@michael_smith

09/17/15

