p.mapper 3

Si possono installare i pacchetti Debian **pmapper-3.2.deb** e **pmapper-base.deb** dal repository

```
deb http://www.pmapper.net/dl/debian binary/
```

Supponiamo che la nostra mappa sia costituita dai layer `confini`, `uso`, `punti`, `linee`, `poligoni` e `homerange`. Una configurazione minima prevede di editare i file:

config/config_default.ini

La prima direttiva da configurare è **pm_config_location**, che indica quale directory di configurazione usare (relativa alla directory **config/**), qui verrano cercati i file `js_config.php` e `php_config.php`:

```
pm_config_location = default
```

Il file **.map** da usare (percorso relativo alla directory specificata in **pm_config_location**) viene indicato nella riga:

```
mapFile = pmapper_demo.map
```

Infine i layer attivi:

```
allGroups = confini, uso, punti, linee, poligoni, homerange
imgFormat = agg_png
;altImgFormatLayers = jpl_wms_global_mosaic, dem
```

config/default/php_config.php

```
$categories['cat_admin']  = array("confini", "uso", "punti", "linee",
"poligoni", "homerange");
$categories['cat_nature'] = array();
$categories['cat_raster'] = array();

$categories_pool['cat_admin']  = array("confini", "uso", "punti", "linee",
"poligoni", "homerange");
$categories_pool['cat_nature'] = array();
$categories_pool['cat_raster'] = array();
```

Un layer che sia definito nel file .map ma che non sia inserito negli array delle categorie sarà sempre visualizzato (deve avere STATUS DEFAULT nel file .map).

config/default/pmapper_demo.map

Il file mappa è il più complesso, per semplificare la fase di debug conviene attivare **error_log** in **/etc/php5/apache2/php.ini**.

Verificare le direttive **WEB.IMAGEPATH** e **WEB.IMAGEURL** che devono indicare la directory in cui

salvare i file temporanei (scrivibile dal server web) e il percorso della stessa rispetto alla DocumentRoot.

Debug

Di grande aiuto è anche provare il file .map fuori dal contesto p.mapper/mapscript, ma direttamente tramite **CGI mapserver**. Questo significa puntare il browser su un URL del tipo:

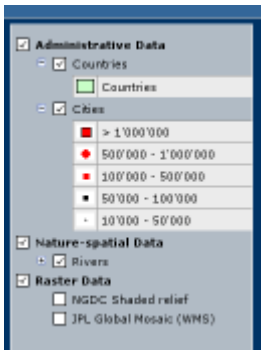
```
http://host/cgi-bin/mapserv?map=/var/www/pmapper-3.2/config/default/pmapper_demo.map&mode=map
```

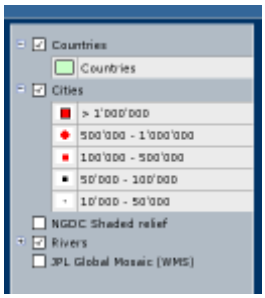
Viene restituita un'immagine con la mappa o l'eventuale messaggio di errore.

ATTENZIONE: Per ottenere la visualizzazione dei layer in **mode=map** CGI è necessario impostare l'attributo **STATUS ON** nella sezione **MAP** e l'attributo **STATUS DEFAULT** in ogni sezione **LAYER**.

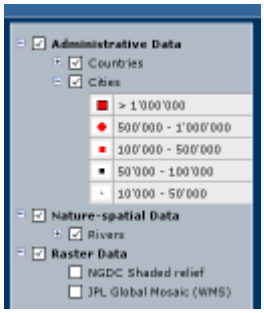
Configurazione TOC (legenda ed elenco layer)

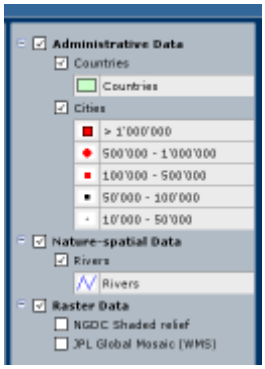
L'elenco dei layer e legenda può essere personalizzato con alcuni parametri in **config/config_default.ini**, ecco alcuni esempi:



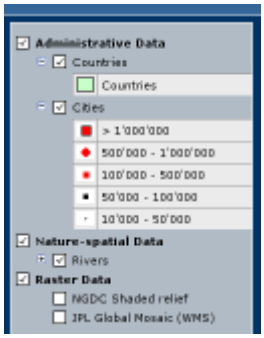


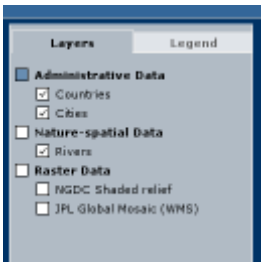
useCategories = {1|0}





grpStyle = {tree|flat}





legStyle = {attached|swap}

Configurazione layout

Modificando i file **`default/php_config.php`** e **`default/js_config.php`** è possibile determinare il logo e varie altre impostazioni della pagina. La pagina viene composta nel file **`map.phtml`**. Alcuni esempi:

```
$pmLogoUrl = "http://www.rigacci.org/";
$pmLogoTitle = "Rigacci.Org Webmapping";
$pmLogo = "images/rigacci_org.png";
```

```
// Top and bottom
Layout.NorthHeight = 35;
Layout.SouthHeight = 35;
Layout.WestWidth   = 0;
Layout.EastWidth   = 240;
```

Modifiche al JavaScript

Tutto il codice JavaScript di pmapper (directory **`javascript/src/`**) viene compattato in alcuni file togliendo spazi e commenti: **`javascript/pm_cjs.js`** e **`javascript/jquery/jquery_merged.js`**. Durante il normale uso di pmapper vengono usati questi file compattatti velocizzando il caricamento della pagina web.

Se si modificano i sorgenti è necessario usare lo script `util/compress_js/compress_js.php` per generare le nuove versioni dei file compattati.

Se si vuole modificare una funzione JavaScript è tuttavia disponibile un meccanismo più sofisticato: è sufficiente definire la stessa funzione (ma con il contenuto modificato) in un file da salvare nella stessa directory della configurazione, ad esempio **`config/default/custom.js`**. Il file JavaScript viene incorporato nella pagina web e la funzione personalizzata sovrascrive quella predefinita.

Per modificare le opzioni JavaScript è disponibile invece il file **`config/default/js_config.php`**. Vedere qui i parametri configurabili. Ad esempio:

```
/** Enbale zoom via mouse wheel as per Google Maps style */
PM.ZoomBox.wheelZoomGoogleStyle = true;

/** Define scale selection list */
```

```
PM.scaleSelectList = ["500", "1.000", "2.000", "5.000", "10.000", "25.000",
"50.000"];
```

Watermark con logo

In realtà si tratta di un layer a tutti gli effetti, disegnato dal MapServer stesso. È un layer speciale con una sola feature (un punto) rappresentato da un solo simbolo (immagine bitmap).

```
SYMBOL
  NAME "logo"
  TYPE PIXMAP
  IMAGE "logo.png"
END

LAYER
  NAME "credits"
  STATUS DEFAULT
  TRANSFORM lr
  TYPE ANNOTATION
  FEATURE
    POINTS
      -70 -60
    END
    TEXT ""
  END
  CLASS
    STYLE
      SYMBOL "logo"
    END
    LABEL
      TYPE BITMAP
      POSITION UL
      COLOR 0 0 0
      BUFFER 5
    END
  END
END
```

Esportazione dati in XLS

Lo strumento query produce una tabella che è esportabile anche in formato XLS. È necessario tuttavia installare i pacchetti Pear **OLE** e **Spreadsheet_Excel_Writer**. Pare che non esistano pacchettizzati Debian. Per vedere se i pacchetti sono già installati ed eventualmente per installarli:

```
pear list
pear install Spreadsheet_Excel_Writer
```

Il comando **pear** dovrebbe attingere automaticamente al repostiory **pear.php.net**. Se il repository non funziona si può scaricare l'archivio del pacchetto da http://pear.php.net/packages.php e poi installarlo con

```
pear install Spreadsheet_Excel_Writer-0.9.1.tgz
```

Personalizzazione stampe

La stampa PDF e HTML viene controllata dal file di configurazione **config/common/print.xml**.

In questo file è possibile ad esempio definire il logo dell'intestazione, il valore predefinito dei checkbox di stampa (*With Overview Map* e *Create PDF Document*).

È possibile ad esempio anche scegliere una dimensione pagina diversa da A4, modificando (sempre in `print.xml`) il dialogbox di stampa con un campo `input` eventualmente `hidden` di nome **papersize** impostato ad A3 o simili. In tal caso bisogna definire nell'XML anche il valore di `print.settings.pdf.format.map`, rispettando anche il map `type` (*normal* o *full*).

Icone legenda

Le icone usate nella legenda sono salvate nella directory **images/legend/** che pertanto deve essere scrivibile dall'utente web server.

In teoria p.mapper dovrebbe accorgersi se il mapfile è stato modificato e in tal caso generare nuovamente tutte le icone per la legenda. Se questo non dovesse avvenire cancellare il file **createimg.log** (o più drasticamente tutte le icone) e ricaricare la pagina.

Join uno a molti

Con lo strumento “seleziona” è possibile effettuare query che eseguono il join tra il dato geografico ed una tabella ad esempio di PostgreSQL. Se il join è di tipo **uno a molti** la tabella risultante mostra tutti i record correlati sulla medesima riga, il risultato è abbastanza illeggibile.

Questa modifica alla funzione **printFields()** (circa alla riga 255 del file **incphp/query/squery.php** di p.mapper 3.2.1) fa sì che ogni record venga stampato su una riga a sé stante:

```
    // if all recors from one2many retrieved (or only one2one) stop loop
    if ($dbloop == $dbresCount) {
        $loop = 0;
    } else {
        $this->qStr .= "], [";
        $this->qStr .= $this->printShapeField($qShape);
    }
```